Санкт–Петербургский государственный университет

Сокол Милена Денисовна

Выпускная квалификационная работа

Распознавание рака груди нейросетевыми методами

Уровень образования: бакалавриат
Направление 02.03.02
«Фундаментальная информатика и информационные технологии»
Основная образовательная программа СВ.5003.2017
«Программирование и информационные технологии»
Профиль «Автоматизация научных исследований»

Научный руководитель:
доцент, кафедра компьютерного моделирования
и многопроцессорных систем,
к.т.н. Гришкин Валерий Михайлович

Рецензент:
профессор, кафедра теории управления и систем,
д.ф. - м.н. Котина Елена Дмитриевна

Санкт-Петербург
2021 г.

Содержание

Введение

По данным на 2018 год рак молочной железы среди женщин самый распространенный вид рака и является самой частой причиной смерти от рака [1]. Поэтому диагноз и лечение на ранних стадиях актуальны для предотвращения прогрессирования заболевания и снижения его заболеваемости.

Для диагностики рака молочной железы обычно используют пальпацию и регулярные осмотры с использованием маммографии или ультразвуковой визуализации. Затем если выявляется высокая вероятность развития злокачественной ткани, врачи делают биопсию молочной железы. Так они гистологически оценивают микроскопическую структуру и элементы ткани. Гистология позволяет различать нормальные ткани, доброкачественные изменения и злокачественные поражения.

Анализ результатов гистологии требует интенсивной работы высококвалифицированных специалистов. Компьютерные системы диагностики, будучи второй системой мнений, могли бы снизить нагрузку на специалистов, что способствовало бы как эффективности диагностики, так и снижению затрат.

Постановка задачи

Самый распространенный тип рака молочной железы — инвазивный рак молочной железы без особого типа (NST) [3]. В некоторых источниках используется старое название инвазивная протоковая карцинома (IDC). Для диагностики рака молочной железы ткань, собранная во время биопсии, обычно окрашивается гематоксилином и эозином. Ткани с инвазивной карциномой демонстрируют искажение архитектуры, а также более высокую плотность и изменчивость ядер, тогда как в нормальной ткани сохраняется архитектура и ядра хорошо организованы (рис. 1). Полученные изображения называют гистологическими снимками или гистологическими изображениями.

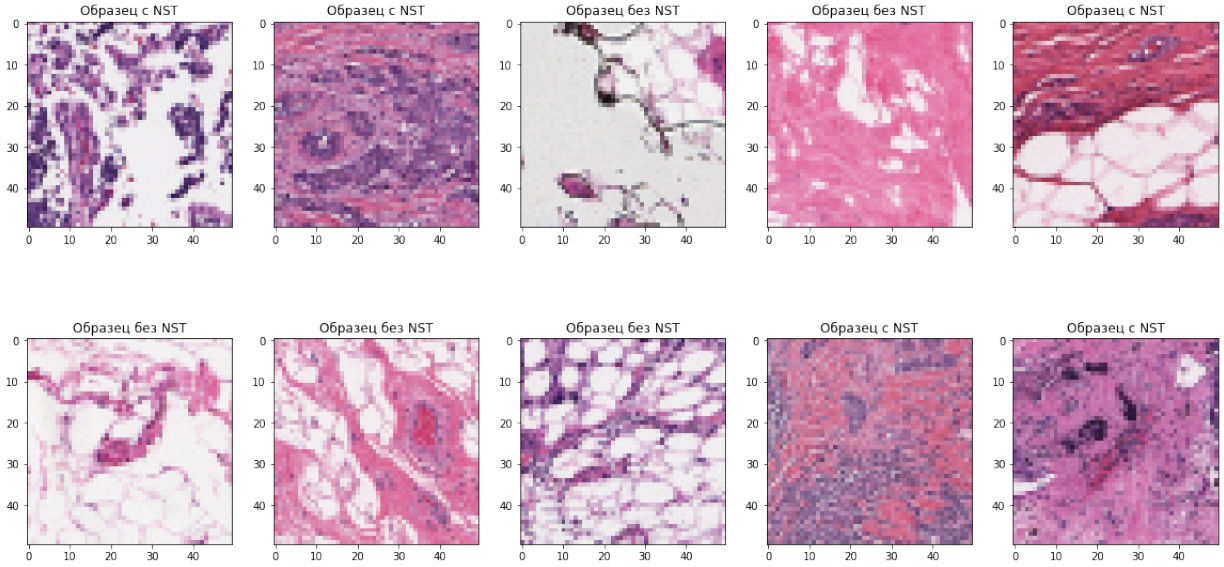

Рис. 1: Гистологические изображения с инвазивным раком молочной железы без особого типа и без него.

Задача, рассматриваемая в этой статье — диагностика инвазивного рака молочной железы без особого типа по гистологическим изображениям. Для решения этой задачи необходимо разработать архитектуры сверточной нейронной сети, соответствующие поставленной задаче, рассмотреть подход для их обучения, провести эксперимент и проанализировать результаты.

Обзор литературы

Во время написания данной работы были использованы книги, статьи и электронные ресурсы для изучения предметной области и исследования существующих методов решения подобных задача.

Для общего понимания тем рака, гистологии, а также определения инвазивного рака молочной железы без особого типа была рассмотрена книга [2], статья [3] и изучены существующие работы по распознаванию рака при помощи сверточных нейронных сетей: [7], [8], [9], [10].

Теоретические знания из области машинного обучения и нейронных сетей получены из лекций К. В. Воронцова [12]. В них он подробно рассказывает теорию обучения по прецедентам, рассматривает влияние l_2 на обучение модели и определяет базовые понятия нейронных сетей. Более углубленное понимание области сверточных нейронных сетей и глубокого обучения, а также ознакомление с реализацией подобных моделей были получены при помощи книги [13], курсов [14] и статей [17], [18]. В источнике [14] были описаны, как классические архитектуры нейронных сетей, не требующие больших мощностей, как LeNet и AlexNet, так и сложные усовершенствованные модели, как VGG. Для налучшего понимания архитектуры LeNet была изучена оригинальная статья [20]. Для изучения альтернативной архитектуры AlexNet была рассмотрена статья [21], где авторы подробно рассказали преимущества своих методов для задачи ImageNet. В этой же статье был подробно описан метод регуляризации Dropout.

Во многих источниках выше был использован метод оптимизации Adam. Для подробного ознакомления с этим методом была использована статья [15]. Для ускорения обучения нейронных сетей был изучен метод батч-нормализации при помощи источников [19] и [14]. Во время реализации моделей нейронных сетей была проанализирована документация [22], в которой были указаны формулы использующихся методов и краткая теория по ним. Для анализа получившихся результатов после проведения эксперименты были рассмотрены различные метрики при помощи электронного ресурса [16] и документации библиотеки, реализующей использованные метрики [23].

Глава 1. Актуальность работы

Увеличение вычислительных мощностей и размеров наборов данных позволило успешно применять сверточные нейронные сети в сфере медицины. Модели глубокого обучения добились отличной производительности в задачах классификации изображений в различных областях. Свёрточные нейронные сети были успешно использованы в таких задачах, как скрининг диабетической ретинопатии [4], прогнозирование заболеваний костей [5], оценка возраста костей [6].

Как было замечены выше, задача разработки компьютерной системы диагностики рака молочной железы актуальна на данный момент. Существуют работы, в которых рассматривается решение этой задачи при помощи сверточных нейронных сетей. В работах [7], [8] сверточные сети способны классифицировать ткани пораженные раком и здоровые ткани. В работах [9], [10] сверточные нейронные сети успешно определяют нормальную ткань, доброкачественное поражение, рак в месте локализации и инвазивный рак груди.

В этой статье рассматривается задача распознавания конкретного типа инвазивной карциномы — инвазивный рак груди без особого типа. Обученные модели не смогут распознать рак другого типа, но они смогут уточнить диагноз после предварительной диагностики врачей или других компьютерных систем диагностики.

Глава 2. Данные для обучения

2.1 Описание данных

Для исследования были использованы данные с платформы Kaggle [11]. В данных содержатся гистологические изображения ткани молочной железы различных пациентов. Каждое изображение было нарезано размером 50x50 пикселей. Изображения были разделены на пять непересекающихся по пациентам директорий, в каждой из которых по 12000 изображений.

2.2 Добавление шума и повороты

Даже после обучения на изображениях из базы данных модели могут некорректно работать с изображениями, которые сделаны в других условиях. На результат может повлиять множество факторов: человеческая ошибка, потеря части данных во время передачи информации или какие-либо искажения из-за проблем с аппаратурой. Необходимо сделать модели более устойчивыми к небольшим изменениям в изображении. Для этого модель будет обучаться как на исходных изображениях, так и на изображениях, к пикселям которых будут добавляться с вероятностью, равной нормальному распределению, значения от 0 до 0.2. Нормальное распределение (распределение Гаусса) задается следующей формулой:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}),$$

где μ — математическое ожидание, а σ — среднеквадратичное отклонение. В работе будет использовано стандартное нормальное распределение, то есть $\mu = 0$, а $\sigma = 1$.

Результат подобного преобразования проиллюстрирован на изображении 2.

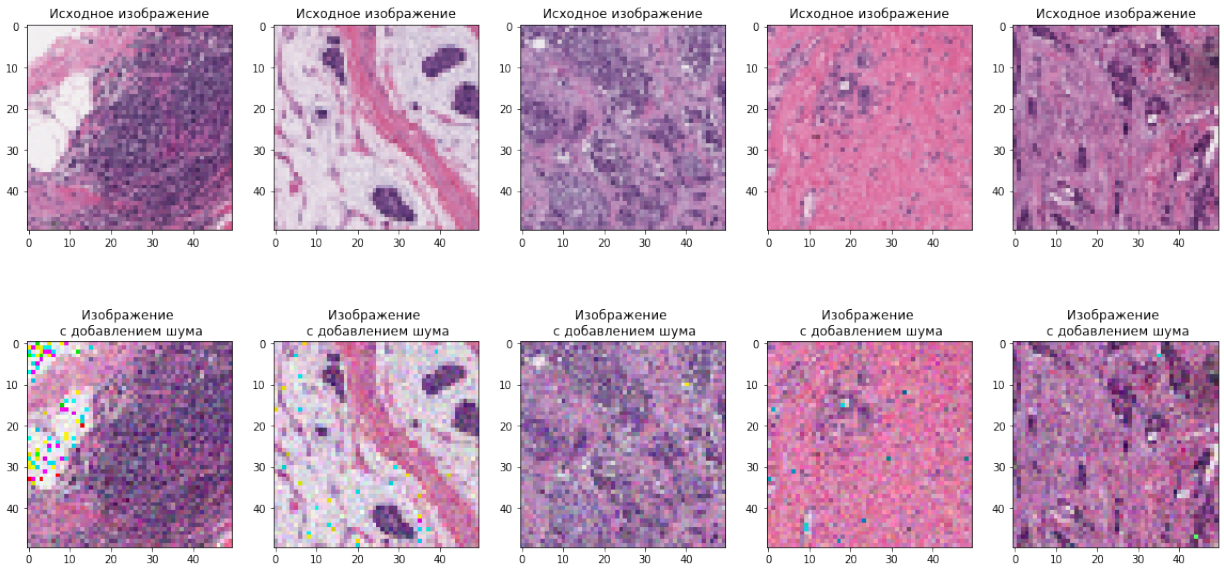

Рис. 2: Добавление шума к изображениям.

Помимо добавления шума, для наилучшей обучаемости нейронной

сети каждый элемент поворачивается на -90, 90 или 180 градуов, и нейронная сеть обучается на новом элементе.

2.3 Кросс-валидация

Как сказано выше, все данные были разделены на пять директорий. В каждой директории 6000 изображений с положительным диагнозом и 6000 — с отрицательным. Первые четыре директории будут использоваться для обучения модели, на данных из пятой директории будет тестироваться обученная модель: эти данные будут называться тестовыми.

Во время обучения необходимо оценивать, насколько хорошо модель работает на незнакомых ей данных. Этот процесс называется валидацией и для него используют валидационные данные. На основе предсказания модели на этих данных можно изменять архитектуру модели или какие-то ее гиперпараметры. Из оставшихся четырех директорий одна будет выделена для валидации, а на трех других модель будет обучаться. Таким образом, тренировочные данные будут составлены из 36000 изображений, валидационные — из 12000 изображений и тестовые также будут составлены из 12000 изображений.

Для исключения случайной зависимости между тренировочными и тестовыми данными, будет использована процедура кросс-валидации. В каждой из четырех директорий, которые выделены на обучение модели, содержится одинаковое количество данных. Для валидации будет выбираться поочередно каждая из папок, а на оставшихся трех будет производиться непосредственно обучение. Таким образом, модель будет обучена четыре раза на разных тренировочных данных. Если во всех случаях результаты на тестовых данных будут удовлетворительны, то можно считать что модель обладает обобщающей способностью.

Глава 3. Нейронные сети

3.1 Задача машинного обучения

Рассмотрим постановку задачи машинного обучения в общем виде. В некоторых источниках ее называют задачей обучения по прецедентам. X — множество объектов, Y — множество ответов, функция y: X $\to$ Y — целевая функция (target function). Даны некоторые элементы множества X и соотвествующие им значения целевой функции:

1. $x_1, ..., x_l \subset X$ — обучающая выборка (training sample);
2. $y_i = y(x_i), i = 1, ..., l$ — известные ответы.

Необходимо найти $\overline{y}$: X $\to$ Y — алгоритм, решающую функцию (decision function), приближающую y на всем множестве X.

3.2 Перцептрон

В 1950-х годах Фрэнк Розенблатт описал базовую для нейронных сетей конструкцию однослойного перцептрона для входных данных x:

$$f(x) = \begin{cases} 1, & \text{if w} \cdot \text{x} + \text{b} > 0 \\ 0, & \text{otherwise} \end{cases}.$$

У этой модели w и b — обучаемые параметры: w, коэффициент скалярного произведения w·x, называется весами (weights), а b — отклонением (bias). Эту модель можно записать в более общем виде:

$$f(x) = g(w \cdot x + b),$$

где

$$g(x) = \begin{cases} 1, & \text{if x} > 0 \\ 0, & \text{otherwise} \end{cases}.$$

В таком виде g называется функцией активации. Конкретно эта функция активации называется единичной ступенькой (step). Существуют и другие функции активации, от выбора которых может измениться скорость и точность обучения модели. Рассмотрим три из них, которые будут использоваться далее:

- логистическая функция активации (Sigmoid):

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

- гиперболичесский тангенс (tanh):

$$tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

- Rectified Linear Units (ReLU):

$$ReLU(x) = \begin{cases} x, & \text{if x} > 0 \\ 0, & \text{otherwise} \end{cases}.$$

3.3 Многослойный перцептрон

Многослойный перцептрон представляет из себя суперпозиции однослойного перцептрона, как, например, представлено на изображении 3. Такая нейронная сеть состоит, как минимум, из трех слоев: входного, промежуточного и выходного.

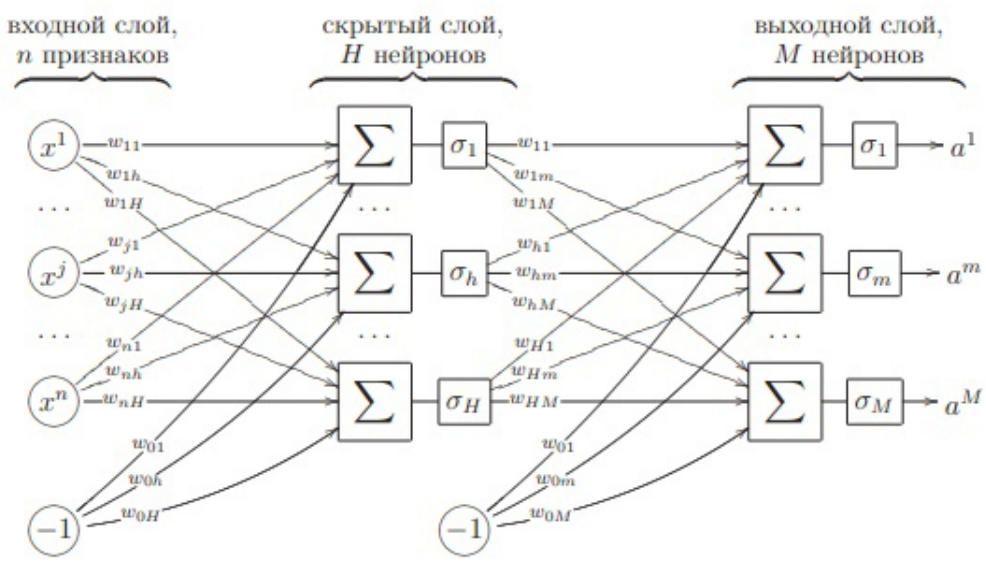

Рис. 3: Архитектура многослойного перцептрона из трех слоев.

Глава 4. Обучение нейронной сети

4.1 Функция ошибки

Нейронная сеть обучается путем минимизации функции ошибки или функции потерь (loss function). Функция потерь является некоторой мерой, по которой можно оценить разницу предсказаний модели от реальных значений целевой функции. Можно сказать, что функция потерь оценивает стоимость ошибки предсказания. В этой работе для обучения будет использоваться функция потерь, называемая кросс-энтропией (cross-entropy) или перекрестной энтропией. Эта функция используется в задаче классификации: для каждого элемента входных данных модель должна определить его класс. Кросс-энтропия определяет близость распределения p значений x, которое задано известными ответами, к распределению q, предсказываемое моделью. Формула этой функции потерь выглядит так:

$$H(p, q) = -\sum_x p(x) \log q(x)$$

4.2 Методы оптимизации

4.2.1 Стохастический градиентный спуск

Как уже было замечено выше, нейронная сеть обучается путем минимизации функции ошибки. Для этого принято использовать метод градиентного спуска. Градиент функции — вектор, направленный в сторону наискорейшего роста функции, его можно найти, как вектор, компонентами которого являются частные производные этой функции.

Так как необходимо найти минимум функции ошибки, то в методе градиентного спуска рассматривается вектор противоположный градиенту, то есть вектор градиента с отрицательным знаком. Метод градиентного спуска — это итерационный метод для оптимизации гладких функций путем движения вдоль вектора градиента. Аргументы этой функции — параметры нейронной сети (например, веса и смещения сверточного слоя этой сети). Обучение нейронной сети с помощью метода градиентного спуска описывается таким алгоритмом:

1. Задаются начальные параметры нейронной сети.

2. На вход сети подаются данные $x_i, ..., x_l$ из обучающей выборки с ответами $y_i, ..., y_l$.

3. Для каждого элемента данных высчитывается результат нейронной сети, функция ошибки и ее градиент.

4. После этого веса нейронной сети изменяются согласно следующему правилу:

$$w = w - \eta \sum_{i=0}^{l} \nabla F_i(w),$$

где w — параметры нейронной сети, η — шаг обучения, который может быть произвольным, $\nabla F_i(w)$ — градиент функции ошибки на данных x_i, y_i, посчитанный на шаге 3.

Для остановки алгоритма высчитывается критерий, называемый метрикой. Когда метрика достигает определенного значения, алгоритм оста-

навливается. В самом простом случае можно использовать значение функции потерь, как метрику. Если метрика не достигает определенного значения, то алгоритм останавливается после определенного количества итераций. В случае стохастического градиентного спуска (stochastic gradient descent — SGD) изменение весов происходит не для всего датасета, а для каждого x_i по-отдельности, где индексы i=0..l выбираются случайным образом.

4.2.2 Adam

Дидерик П. Кингма и Джимми Ба в 2015 году опубликовали статью [15], в которой представили метод оптимизации Adam. Этот метод схож с градиентным спуском, но у него есть существенные преимущества. Для каждого параметра нейронной сети Adam настраивает свой шаг обучения. Это делается при помощи экспоненциального скользящего среднего (exponential moving average — EMA):

$$EMA(t, f, \alpha) = \alpha * f(t) + (1 - \alpha) * EMA(t - 1, f, \alpha).$$

В формуле выше t — это момент времени, а f(t) — функция, для которой рассматривается EMA, α — параметр, настраиваемый пользователем. Adam выполняет те же действия, что и алгоритм SGD, за исключением функции изменения весов. Для алгоритма Adam она выглядит так:

$$w(t + 1) = w(t) - \alpha \frac{EMA(t, \nabla F(t), \beta_1)}{\sqrt{EMA(t, \nabla F(t)^2, \beta_2)} + \epsilon}.$$

w — параметры нейронной сети, α, β_1, β_2 — параметры, которые выбираются пользователем, $\nabla F(t)$ — градиент функции ошибки на итерации t, ϵ — очень маленькое число, введеное для того, чтобы исключить деление на ноль.

4.3 Метрики

В этой работе будут использованы следующие метрики: точность, F1 и ROC-кривая.

Точность — доля (процент) элементов из обучающей выборки, на которых нейронная сеть выдала правильные ответы.

В случае бинарной классификации ответы разделяются на положительные (1) и отрицательные (0). Предсказания могут быть истинными и ложными, тогда их можно разделить на четыре типа:

- истинно-положительные(true positive) — модель верно определила элемент как положительный (количество таких элементов обозначим далее TP);
- истинно-отрицательные(true negative) — модель верно определила элемент как отрицательный (количество таких элементов обозначим далее TN);
- ложно-положительные(false positive) — модель неверно определила элемент как положительный (количество таких элементов обозначим далее FP);
- ложно-отрицательные(false negative) — модель неверно определила элемент как отрицательный (количество таких элементов обозначим далее FN).

Для определения ROC-кривой и F1-меры нужно ввести дополнительные метрики:

1. Чувствительность (sensitivity)

$$Sen = \frac{TP}{TP + FN};$$

2. Специфичность (specificity)

$$Spe = \frac{TN}{TN + FP};$$

3. Точность (precision)

$$Pre = \frac{TP}{TP + FP}.$$

Если при оценке обучения есть необходимость учитывать и чувствительность, и точность, то можно использовать метрику F1. Она представляет из себя их гармоническое среднее:

$$F1 = 2 * \frac{Pre \cdot Sen}{Pre + Sen}.$$

Нейронная сеть с логистической функцией активации на выходе выдает не значение класса, а вероятность принадлежности к классу. Для получения значения класса вводится порог. Если вероятность ниже порога, то считается, что элемент не принадлежит классу, если выше — принадлежит.

Если менять такой порог, то будут меняться значения TN, FN, TP и FP, а значит чувствительность и специфичность. Если меняющиеся значения 1 - Spe поместить на ось X, а значения Sen — на ось Y, то такой график и будет называться ROC-кривой (receiver operating characteristic). Площадь под ней называется AUC-ROC (area under curve), она же является численной метрикой для оценки качества предсказания нейронной сети. AUC-ROC может принимать значения от 0 до 1: чем значение больше, тем модель работает лучше.

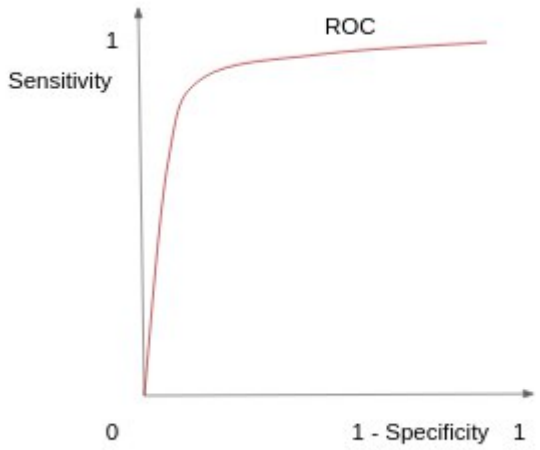

Рис. 4: Пример ROC-кривой.

4.4 Регуляризация

Нейронная сеть обучается на конкретной выборке. Иногда происходит так, что она запоминает признаки, которые присущи объектам только на обучающей выборке, и тогда модель ошибается на других данных. Это называется переобучением. Один из спобов предотвратить такую ситуацию — регуляризация. Регуляризация — это добавление некоторых дополнительных ограничений к условию задачи. В этой работе будет использоваться l_2-регуляризация (ridge regression), которая накладывает штраф за слишком большую норму весов нейронной сети. Этот штраф накладывается на функцию потерь таким образом:

$$l_2 = loss + \lambda \sum_i weight_i^2, \tag{1}$$

где loss — это выбранная функция потерь, $weight_i$ — i-ый параметр нейронной сети, λ — параметр, настраиваемый пользователем. В этом случае минимизируется не выбранная функция сама по себе, а функция l_2.

Глава 5. Сверточные нейронные сети

Концепция свёрточных нейронных сетей (CNN - Convolutional Neural Networks) была предложена Яном Лекуном в 1988 году, вдохновленным работами нобелевских лауреатов в области медицины. Идея заключается в чередовании сверточных слоев, субдискретизирующих слоев и наличии полносвязных слоев на выходе.

5.1 Свёрточный слой (Convolution)

Слой свёртки — ключевой слой для модели свёрточной нейронной сети, состоящий из нескольких фильтров, у каждого из которых есть ядро со своими весовыми коэффициентами (небольшая матрица). Изначально весовые коэффициенты ядра свёртки неизвестны и устанавливаются в процессе обучения. Выходом сверточного слоя являются результаты скалярного произведения фрагментов входного вектора и ядра свертки, таким обра-

зом пиксели изображения заменяются взвешенной суммой себя и соседних пикселей.

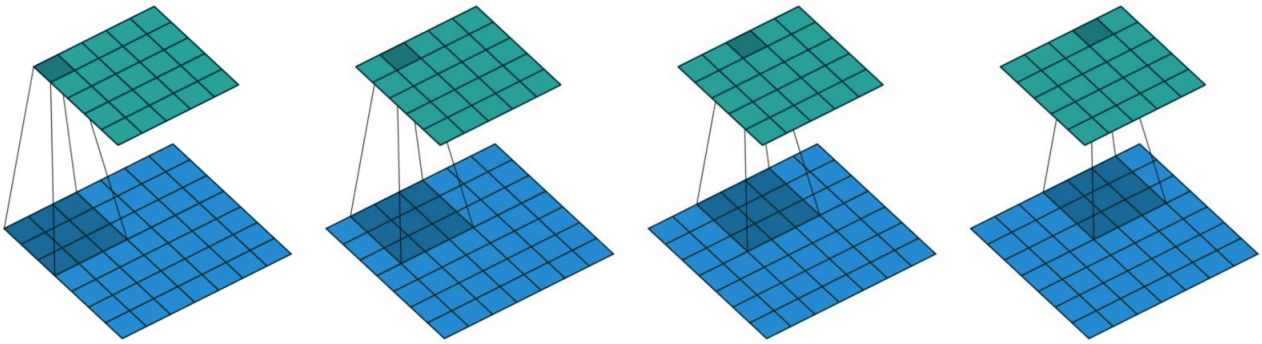

Рис. 5: Визуализация четырех шагов операции свертки: синяя матрица — оригинальное изображение, серая матрица поверх синей — перемещающееся ядро свертки, зеленая — результат операции свертки.

Подобное скалярное произведение называют операцией свертки:

$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \times I_{x+i-1,y+j-1},$$

где I — исходное изображение, K — ядро свертки (матрица размерности $h \times w$), x и y — координаты результата операции.

Для свёрточного слоя существуют гиперпараметры:

- Глубина (depth) — количество признаков (фильтров), получаемых в результате работы нейронной сети. У каждого признака свёрточной сети весовые коэффициенты ядра отличаются;
- Высота (height) и ширина (width) ядер;
- Шаг (stride) — количество пикселей, на которое смещается ядро на каждом шаге при вычислении следующего пикселя результирующего изображения. Обычно его принимают равным 1, и чем больше его значение, тем меньше размер выходного изображения;
- Отступ (padding, zero-padding) — операция дополнения исходного изображения нулями. Свертка любым ядром размерности более, чем 1x1

уменьшит размер выходного изображения. Так как в общем случае желательно сохранять размер исходного изображения, рисунок дополняется нулями по краям.

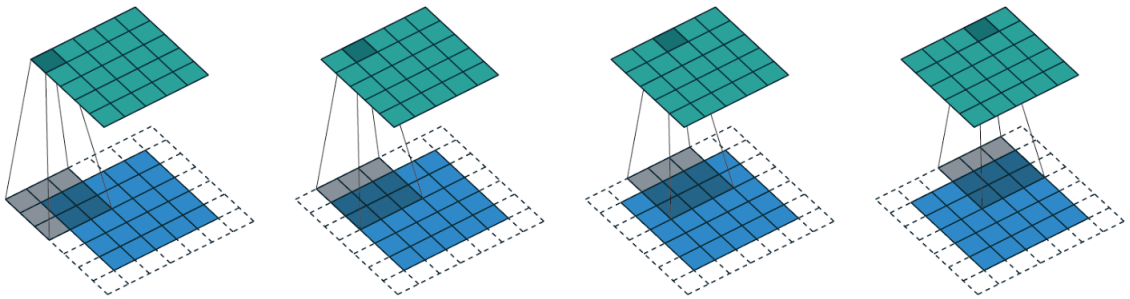

Рис. 6: Визуализация четырех шагов операции свертки с padding = 1.

5.2 Субдискретизирующий слой (Subsampling)

Операция субдискретизации (пулинга) позволяет уменьшить входные данные. Аналогично сверточному слою в субдискретизирующем задается размер ядра, которое с определенным шагом смещается по входным данным и заменяет покрывающиеся пиксели на один элемент. Обычно выбирается максимальный (max-pooling), но иногда берут минимальный или среднее.

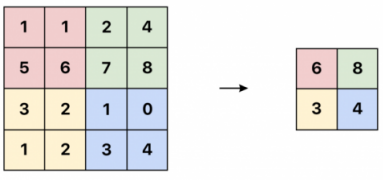

Рис. 7: Визуализация операции max-pooling с размером ядра 2.

5.3 Полносвязный слой (Full connection)

Результаты работы предыдущих слоев обычно переводят в линейный вектор признаков, который подается на вход полносвязному слою. В нем происходит линейное преобразование вектора признаков x весовой матрицей W и вектором смещения b, после чего применяется нелинейная функция активации σ:

$$\sigma(W * x + b).$$

Несколько полносвязных слоев образуют многослойный перцептрон.

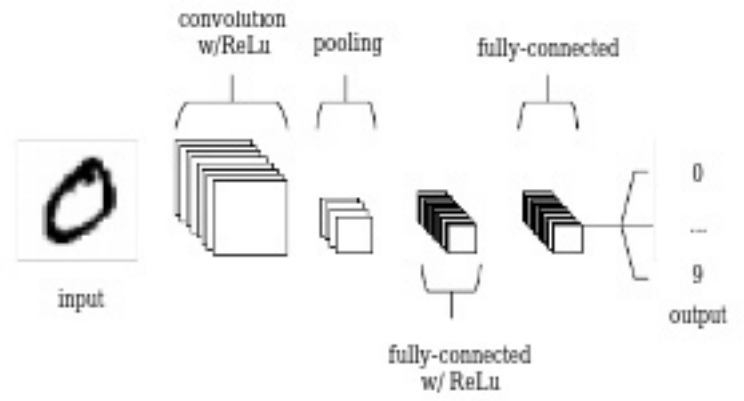

Рис. 8: Пример архитектуры CNN.

5.4 Батч-нормализация (Batch Normalization)

Во время обучения нейронной сети для эффективного использования графического процессора данные делят на равные части — батчи. В процессе обучения распределение данных может меняться на разных батчах, а также из-за изменяющихся весов выходные данные слоев могут тоже иметь разные распределения. Для решения этой проблемы применяют батч-нормализацию. Суть этого метода состоит в том, что перед выбранным слоем батч данных нормализуют так, чтобы математическое ожидание распределения данных равнялось нулю, а дисперсия — единице. Зная математическое ожидание распределения данных μ и дисперсию σ, батч-

нормализацию можно найти в таком виде:

$$\overline{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}},$$

где ϵ — очень маленькое число, использованное для исключения возможности деления на ноль.

После вычисления нормализованных данных $\overline{x}$ для эффективного обучения нейронной сети применяются обучаемые параметры β и γ:

$$\widehat{x} = \gamma\overline{x} + \beta.$$

Глава 6. Преимущества CNN для данной задачи

Во время анализа гистологических снимков специалисты исследуют архитектуру изображений. Зависимость искажения архитектуры гистологических изображений от диагноза позволяет предположить, что свёрточные нейронные сети, способные сохранять информацию о взаимном расположении пикселей, будут эффективны в решении проблемы распознавания NST.

Преимущество CNN в том, что они выделяют признаки непосредственно из изображений, в отличие от традиционных методов, где признаки приходится выбирать вручную. Изучение гистологических снимков нейронными сетями в каком-то смысле схоже с тем, как патологоанатомы рассматривают ткани по фрагментам через разные увеличивающие стекла.

Глава 7. Архитектуры сверточных нейронных сетей

Существует множество известных архитектур для сверточных сетей. В этой работе будут использованы две из них: LeNet и AlexNet. Эти архитектуры проверены временем и использовались в решении множества задач. Их преимущество в том, что они не требуют больших ресурсов для обучения и использования, а их результаты достаточно интерпретируемы.

7.1 LeNet

LeNet — это классическая архитектура сверточной нейронной сети, предложенная Яном Лекуном [20]. Такая нейронная сеть состоит из чередования сверточных слоев с гиперболическим тангенсом на выходе и усредняющих слоев дискретизации. После сверточных слоев идут два полносвязных также с гиперболическим тангенсом на выходе. И третий полносвязный слой возвращает столько элементов, сколько классов нужно предсказать. После чего, из этих элементов получают вероятности слоем активации Softmax. Softmax — это обобщенная логистическая функция, на случай если количество классов больше двух. Ян Лекун рассматривал задачу классификации рукописных цифр на 10 классов, но в поставленной в этой статье задаче их будет всего два, а значит можно использовать логистическую функцию Sigmoid.

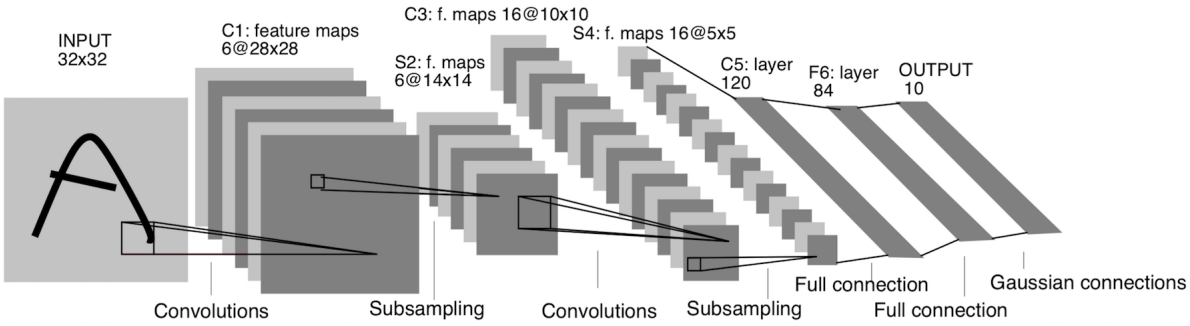

Рис. 9: Архитектура LeNet для классификации MNIST.

7.2 AlexNet

В 2012 году Алексей Крижевский опубликовал статью с описанием модели, схожей с предыдущей, но с очень значимыми изменениями, благодаря которым эта модель победила в соревновании ImageNet [21]. Как видно на изображении 10, в этой модели больше сверточных слоев, а у каждого слоя больше весов, так как размер фильтров больше. На выходе сверточных слоев авторы модели использовали не гиперболический тангенс, а функцию ReLU. Благодаря этому модель обучается быстрее. В субдискретизирующих слоях авторы статьи не усредняли значения (average-pooling),

а брали максимальное (max-pooling). Но для слоев субдескритизации было введено еще одно важное изменение: если в LeNet шаг операции субдескритизации был равен размеру ядра, то есть ядро перемещалось по изображению, каждый раз рассматривая новый фрагмент, то в AlexNet ядро перемещалось по изображению с шагом меньше, чем длина ядра, то есть в нескольких итерациях могли рассматриваться одни и те же фрагменты изображения. Таким образом на разных итерациях ядра как бы перекрывали друг друга. Такой подход позволяет избежать потери важного свойства.

Авторы статьи добавили к модели метод регуляризации Dropout, который позволяет избежать переобучения, каждый раз модифицируя архитектуру нейронной сети. Этот метод применяется после какого-либо слоя, с определенной вероятностью исключая из обучения нейроны этого слоя. Этот подход позволяет исключить взаимное влияние нейронов. В каком-то смысле это схоже с удалением некоторых пикселей изображения для исключения случайных связей, но удаляются не пиксели, а информация, которую была получена внутри нейронной сети. В модели AlexNet этот метод применяется на двух первых полносвязных слоях. В статье замечено, что без этого метода модель достаточно быстро переобучается.

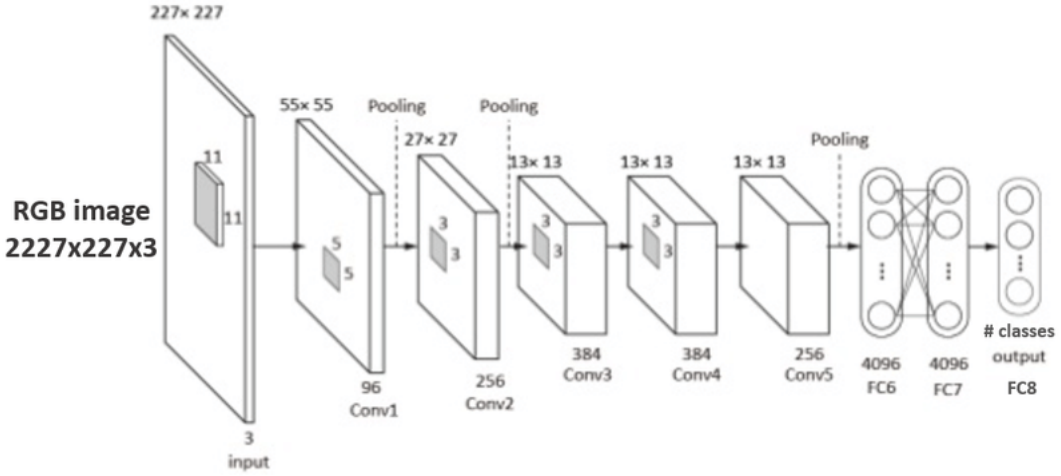

Рис. 10: Архитектура AlexNet.

Глава 8. Результаты

8.1 Использованные инструменты

Для реализация решения задачи был использован сервис Google Colab, в основе которого платформа Jupyter Notebook. Код решения представлен на языке Python. Для составления тренировочных, валидационных и тестовых данных была использована библиотека Torchvision, для визуализации данных, процесса обучения и результатов эксперимента была использована библиотека Matplotlib, для реализации нейронных сетей и их обучения использовалась библиотека PyTorch, для некоторых математических вычислений была задействована библиотека NumPy. Ссылка на документ с кодом эксперимента — в приложениях к данной статье. В документе подробно расписан весь процесс реализации эксперимента.

8.2 Реализованные архитектуры

В процессе работы было реализовано две модели: NST-LeNet и NST-AlexNet, разработанные по аналогии с архитектурами LeNet и AlexNet соответственно. Между некоторыми слоями дополнительно был добавлен метод батч-нормализации.

NST-LeNet описана в таблице 1. В первом столбце таблицы — номер слоя, во втором столбце — тип слоя сверточной нейронной сети, в третьем — размер ядра слоя, если у слоя оно есть, в четвертом — шаг, с которым ядро перемещается по входному изображению, в последнем стобце — функция активации, используемая после всех преобразований. В слое Flatten изображение после сверточных слоев растягивается в вектор.

Архитектура NST-AlexNet описана в таблице 2 по аналогии с таблицей 1. Шестым столбцом в нее добавлен отступ (padding) в случае если он есть у слоя. В архитектуре NST-LeNet отступ не использовался. NST-AlexNet использует слой Dropout. В таблице он указан с параметром в скобках — вероятностью, с которой нейрон будет исключен после работы предыдущего слоя. Достаточно большой параметр, 0.7, был выбран для предотвращения переобучения модели.

Таблица 1: Архитектура NST-LeNet

Слой	Тип	Кол-во признаков	Размер ядра	Шаг	Активация
0	Исходное изображение	3	—	—	—
1	Сверточный	18	4	1	Tanh
2	Average Pooling	18	2	2	—
3	Батч-нормализация	18	—	—	—
4	Сверточный	36	4	1	Tanh
5	Average Pooling	36	2	2	—
6	Батч-нормализация	36	—	—	—
7	Сверточный	48	4	1	Tanh
8	Average Pooling	48	2	2	—
9	Батч-нормализация	18	—	—	—
10	Сверточный	54	3	1	Tanh
11	Flatten	54 * 1 * 1	—	—	—
12	Полносвязный	32	—	—	Tanh
13	Полносвязный	16	—	—	Tanh
14	Полносвязный	2	—	—	Sigmoid

Таблица 2: Архитектура NST-AlexNet

Слой	Тип	Кол-во признаков	Размер ядра	Шаг	Отступ	Активация
0	Исходное изображение	3	—	—	—	—
1	Сверточный	16	5	2	4	ReLU
2	Батч-нормализация	16	—	—	—	—
3	Сверточный	32	4	1	2	ReLU
4	Max Pooling	32	2	2	—	—
5	Батч-нормализация	32	—	—	—	—

6	Сверточный	64	3	1	2	ReLU
4	Max Pooling	64	2	2	—	—
7	Батч-нормализация	64	—	—	—	—
8	Сверточный	64	3	1	1	ReLU
7	Батч-нормализация	64	—	—	—	—
8	Сверточный	32	3	1	1	ReLU
9	Max Pooling	32	2	2	—	—
11	Flatten	32 * 4 * 4	—	—	—	—
10	Dropout(0.7)	32 * 4 * 4	—	—	—	—
12	Полносвязный	32	—	—	—	ReLU
13	Dropout(0.7)	32	—	—	—	—
14	Полносвязный	16	—	—	—	ReLU
15	Полносвязный	2	—	—	—	Sigmoid

8.3 Результаты экспериментов

Модели были обучены при помощи метода оптимизации Adam и l_2 регуляризации на исходных данных, данных с шумами и повернутых данных. Результаты обучения анализировались при помощи кросс-валидации.

Нейронная сеть на выходе выдает вектор состоящий из вероятности отрицательного ответа и вероятности положительного ответа. Как результат предсказания для метрик Accuracy и F1 рассматривается ответ, соответствующий наибольшей вероятности. Для метрики AUC-ROC используются вероятности положительного ответа.

Результаты обучения указаны в таблицах 3, 4, 5. В них стобцы 0, 1, 2 и 3 обозначают обучение, где данные лежащие в директории под этим номером рассматриваются, как валидационные. Остальные три директории составляют тренировочные данные. Метрики в таблицах получены на тестовых данных.

Таблица 3: Значения метрики Accuracy на кросс-валидации.

Модель	0	1	2	3
NST-LeNet	0.75	0.86	0.83	0.80
NST-AlexNet	0.79	0.87	0.90	0.87

Таблица 4: Значения метрики F1 на кросс-валидации.

Модель	0	1	2	3
NST-LeNet	0.75	0.86	0.83	0.80
NST-AlexNet	0.79	0.87	0.90	0.87

Таблица 5: Значения метрики AUC-ROC на кросс-валидации.

Модель	0	1	2	3
NST-LeNet	0.81	0.95	0.92	0.89
NST-AlexNet	0.94	0.96	0.96	0.95

8.4 Визуализация процессов эксперимента

Для анализа обучения модели были составлены графики изменения функции потерь на тренировочных данных синим цветом и на валидационных — красным. На рис 11 изображен процесс изменения функции потерь во время обучения модели NST-LeNet. Аналогичный график на рис. соответствует изменению метрики точности 12.

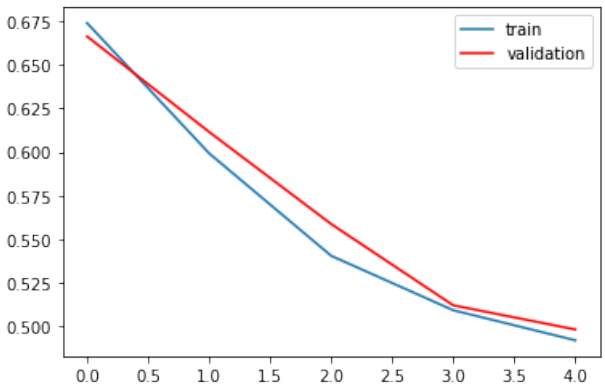

Рис. 11: Изменение функции потерь во время обучения.

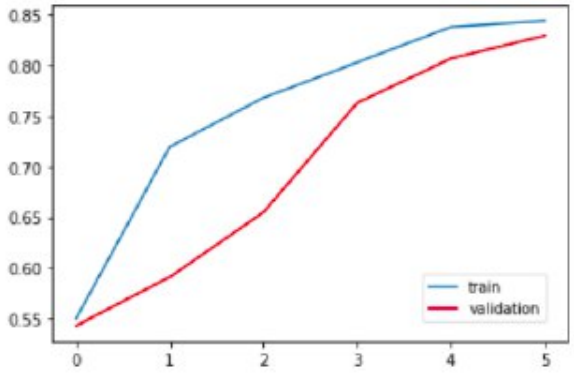

Рис. 12: Изменение точности во время обучения.

По этим изображениям можно оценить, что модель обучалась достаточно быстро и переобучения не было.

После обучения для интерпретации обучения первого слоя нейронных сетей были рассмотрены их выходные изображения. На рис. 13 изображены данные с отрицательным диагнозом, на рис. 14, 15 выходы первого слоя для моделей NST-LeNet и NST-AlexNet на этих данных.

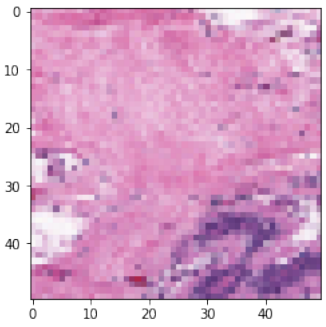

Рис. 13: Изображение ткани без NST.

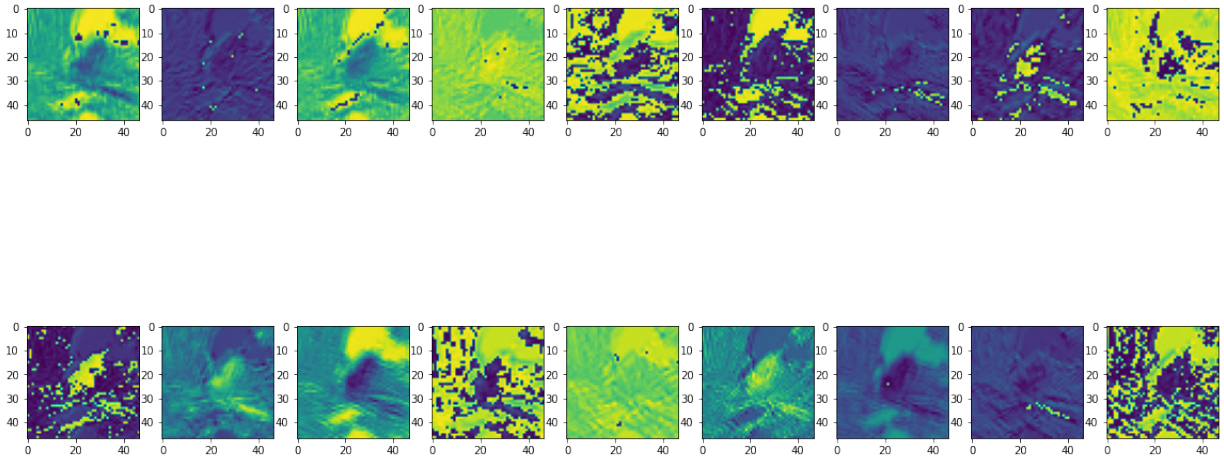

Рис. 14: Выходы первого слоя архитектуры NST-LeNet ткани с рис. 13.

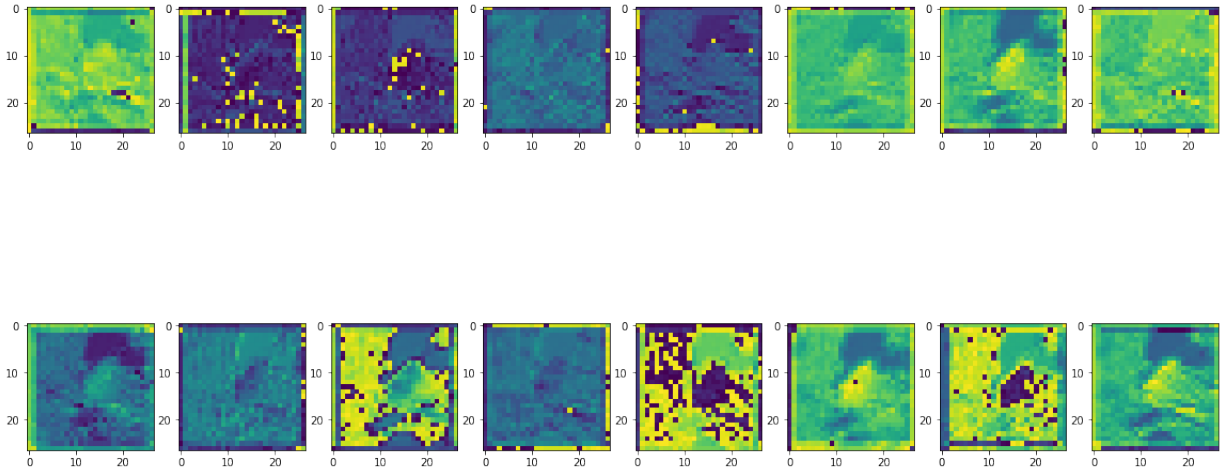

Рис. 15: Выходы первого слоя архитектуры NST-AlexNet ткани с рис. 13.

На рис. 16 изображены данные с отрицательным диагнозом и на рис. 17, 18 выходы первого слоя для моделей NST-LeNet и NST-AlexNet на этих данных.

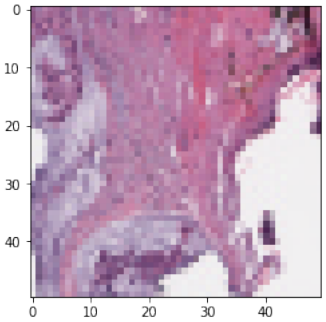

Рис. 16: Изображение ткани с NST.

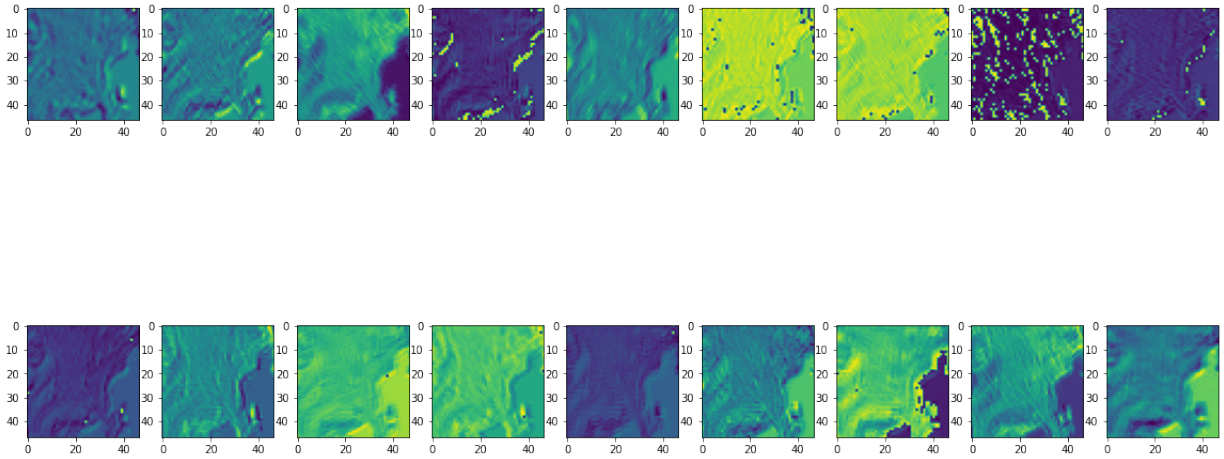

Рис. 17: Выходы первого слоя архитектуры NST-LeNet ткани с рис. 16.

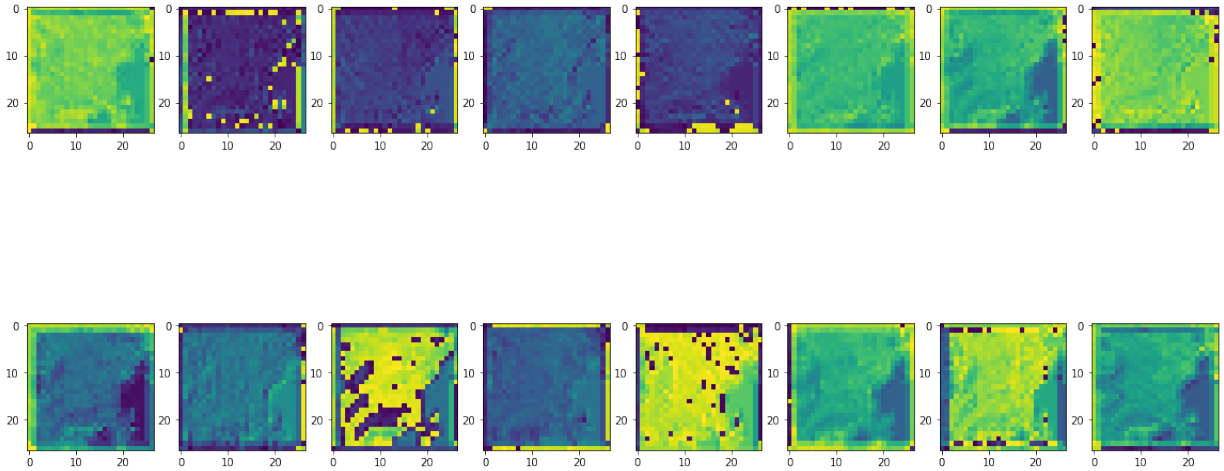

Рис. 18: Выходы первого слоя архитектуры NST-AlexNet ткани с рис. 16.

Анализируя получившиеся изображения, можно сделать вывод, что модели действительно находит особенности архитектуры изображения. На выводах модели AlexNet также можно увидеть отступ, заданный на первом слое.

Глава 9. Выводы

В результате эксперимента NST-AlexNet показала результаты лучше, чем NST-LeNet. Это можно объяснить теми преимуществами AlexNet перед LeNet, которые описывались в разделе 7.2. Как основные преимущества можно выделить большее количество параметров и метод Dropout.

На кросс-валидации заметно, что результаты обучения в случае, где валидационные данные лежат в нулевой директории, значительно хуже. Возможно, в этой директории лежат специфичные замеры существенно отличающиеся от остальных.

Тем не менее, результаты обучения на обеих моделях удовлетворительные. На основании результатов эксперимента можно сделать вывод, что сверточные нейронные сети способны распознавать инвазивную карциному молочной железы без особого типа.

Заключение

Целью данной работы была разработка моделей, которые были бы способны распознавать инвазивный рак груди без особого типа, как вторая система мнений для специалистов. В процессе работы были рассмотрены существующие решения и сделан вывод, что разрабатываемые модели могли бы уточнить диагноз после работы существующих моделей, способных распозновать инвазивные карциомы.

Далее были рассмотрены классические модели, как AlexNet и LeNet и сделан вывод, что как модели распознающие паттерны в изображениях и не требующие больших ресурсов для обучения и использования, они соответствуют поставленной задаче. По аналогии с моделями AlexNet и LeNet были разработаны архитектуры NST-AlexNet и NST-LeNet соответственно.

Для обучения этих нейронных сетей был использован метод оптимизации Adam, способный подбирать шаг обучения для каждого параметра индивидуально. Для регуляризации функции ошибки была использвана l_2 регуляризация в обоих моделях и метод Dropout в модели NST-AlexNet. Также для увеличения способности моделей распознавать данные с искажениями во время обучения к данным добавлялся шум. Для ускорения обучения моделей перед некоторыми слоями использовался слой батч-нормализации.

После обучения были проанализированы метрики, полученные на кросс-валидации. Из них следовало, что модели способны находить признаки, характерные для тканей с NST. Для интерпретации работы первого сверточного слоя моделей были рассмотрены выходы этого слоя на изображении с NST и без него. По изображениям выходов был сделан вывод, что первый слой моделей способен распознавать архитектуру изображения.

Полученные результаты удовлетворительны, но есть потенциал к более успешному распознаванию инвазивного рака груди без особого типа. Это можно сделать путем рассмотрения более затратных по ресурсам моделей, а также при помощи консультаций со специалистами.

Список литературы

[1] J. Ferlay, M. Colombet, I. Soerjomataram et al «Estimating the global cancer incidence and mortality in 2018: GLOBOCAN sources and methods», 2018 // URL: https://onlinelibrary.wiley.com/doi/epdf/10.1002/ijc.31937.

[2] Блохин Н. Н., Петерсон Б.Б,«Клиническая онкология», 1979.

[3] Hans-Peter Sinn, Hans Kreipe,«A Brief Overview of the WHO Classification of Breast Tumors, 4th Edition, Focusing on Issues and Updates from the 3rd Edition», 2013 // URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3683948/.

[4] Alexander Rakhlin, «Diabetic Retinopathy detection through integration of Deep Learning classification framework», bioRxiv, 225508, 2017.

[5] Aleksei Tiulpin, Jérôme Thevenot, Esa Rahtu et al, «Automatic Knee Osteoarthritis Diagnosis from Plain Radiographs: A Deep LearningBased Approach», Scientific Reports, 8, 1727, 2018.

[6] Vladimir Iglovikov, Alexander Rakhlin, Alexandr A. Kalinin et al, «Pediatric Bone Age Assessment Using Deep Convolutional Neural Networks», arXiv preprint arXiv:1712.05053, 2017.

[7] Angel Cruz-Roa, Ajay Basavanhally, Fabio Gonz´alez et al, «Automatic detection of invasive ductal carcinoma in whole slide images with Convolutional Neural Networks», 2014

[8] Fabio Alexandre Spanhol, Luiz S. Oliveira, Caroline Petitjean and Laurent Heutte, «Breast Cancer Histopathological Image Classification using Convolutional Neural Networks», 2016

[9] Teresa Araújo, Guilherme Aresta, Eduardo Castro et al, «Classification of breast cancer histology images using Convolutional Neural Networks», 2017 // URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5453426/

[10] Alexander Rakhlin, Alexey Shvets, Vladimir Iglovikov et al, «Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis», 2018 // URL: https://arxiv.org/pdf/1802.00752.pdf

[11] Breast Histopathology Images [Электронный ресурс] URL: https://www.kaggle.com/paultimothymooney/breast-histopathology-images (дата посещения 20.05.2021)

[12] К. В. Воронцов, «Математические методы обучения по прецедентам», 2011

[13] Николенко С., Кадурин А., Архангельская Е., «Глубокое обучение. Погружение в мир нейронных сетей», 2019

[14] Samsung Research Russia Open Education, Курс «Нейронные сети и компьютерное зрение»[Электронный ресурс] URL: https://stepik.org/course/50352/ (дата посещения 20.05.2021)

[15] Diederik P. Kingma, Jimmy Ba, «Adam: A Method for Stochastic Optimization», 2014

[16] Confusion matrix and other metrics in machine learning [Электронный ресурс] URL: https://medium.com/hugo-ferreiras-blog/confusion-matrix-and-other-metrics-in-machine-learning-894688cb1c0a (дата посещения 30.05.2021)

[17] Vincent Dumoulin, Francesco Visin, «A guide to convolution arithmetic for deep learning», 2018 // URL: https://arxiv.org/pdf/1603.07285v2.pdf

[18] Keiron O'Shea, Ryan Nash, «An Introduction to Convolutional Neural Networks», 2015

[19] A Gentle Introduction to Batch Normalization for Deep Neural Networks [Электронный ресурс] URL: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/ https://stepik.org/course/50352/ (дата посещения 20.05.2021)

[20] Yann Lecun, Y. Bengio «Convolutional Networks for Images, Speech, and Time-Series», 1997

[21] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks», 2012

[22] PYTORCH DOCUMENTATION [Электронный ресурс] URL: https://pytorch.org/docs/stable/index.html (дата посещения 20.05.2021)

[23] TORCHMETRICS DOCUMENTATION[Электронный ресурс] URL: https://torchmetrics.readthedocs.io/en/latest/ (дата посещения 20.05.2021)

Приложения

Программная реализация и веса моделей доступны по ссылке: https://github.
cancer-detection-using-neural-networks.