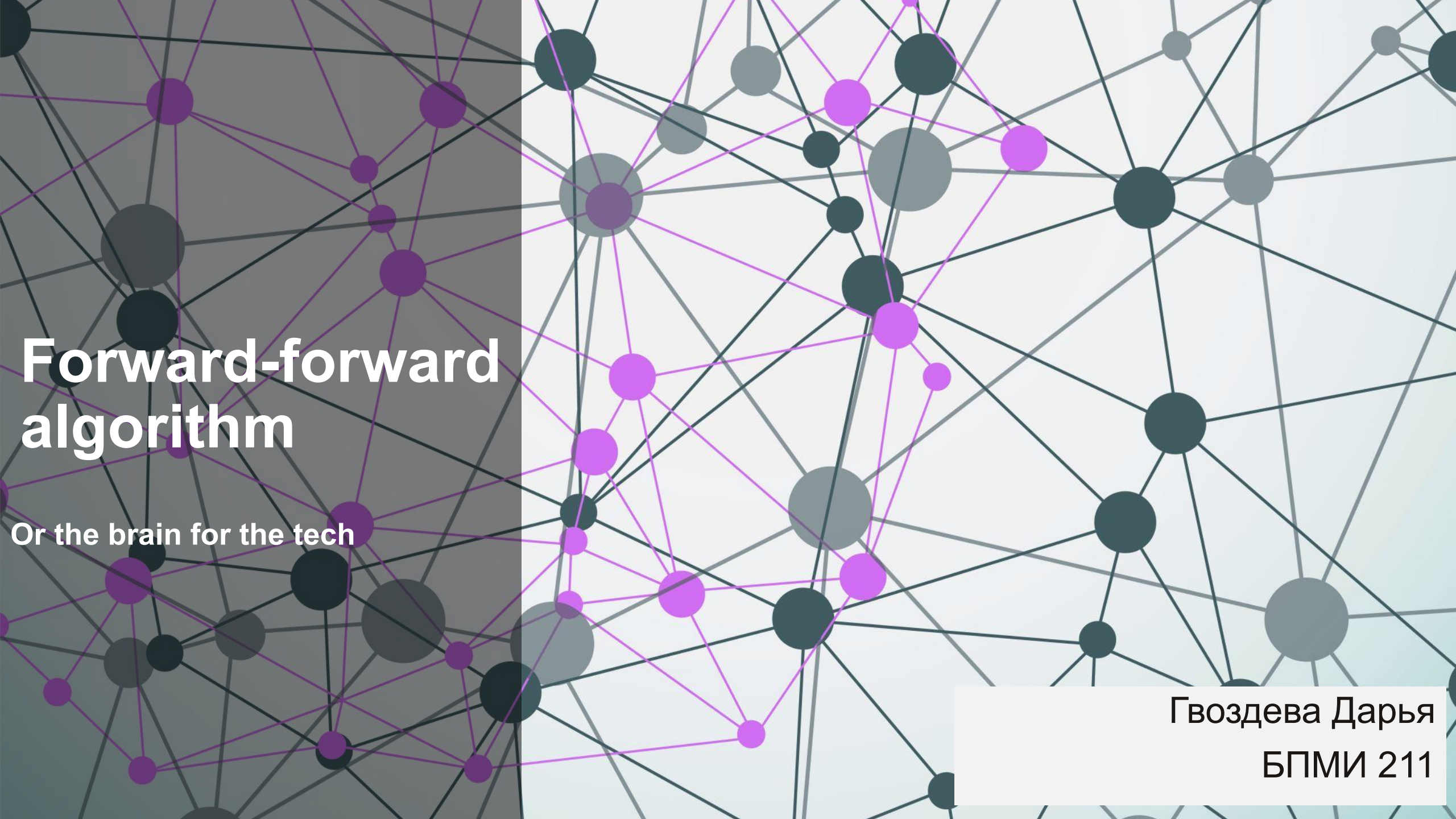


Forward-forward algorithm

Or the brain for the tech

Гвоздева Дарья
БПМИ 211

Что не так с backpropagation (BP)

Эффективность SGD подтвердилась на практике, возник большой интерес к BP. Обнаружилось следующее:

- мозг не реализует свою работу аналогично
- для BP необходимо четко понимать, как мы вычисляли forward pass, что не всегда возможно
- можно было бы прибегнуть к обучению с подкреплением - высокие дисперсии мешают

Что не так с backpropagation (BP)

Эффективность SGD подтвердилась на практике, возник большой интерес к BP. Обнаружилось следующее:

- мозг не реализует свою работу аналогично
- для BP необходимо четко понимать, как мы вычисляли forward pass, что не всегда возможно
- можно было бы прибегнуть к обучению с подкреплением - высокие дисперсии мешают

Цель: показать, что нейронные сети с неизвестными нелинейностями не нуждаются в использовании обучения с подкреплением.

Алгоритм Forward-Forward (FF)

Плюсы:

- по скорости сравним с backpropagation
- его можно использовать, когда точные детали вычислений forward pass неизвестны
- обучается без сохранения нейронных действий или остановки

Минусы:

- алгоритм forward-forward несколько медленнее, чем backpropagation
- не так хорошо обобщает на некоторых игрушечных задачах
- в исследовании больших моделей, обученных на больших данных, BP все равно выигрывает

Так что же такое Forward Forward алгоритм

Два прохода: положительный (с, *вау,* реальными данными) и отрицательный (со сгенерированными данными).

Обращаем внимание на goodness модели. Цель обучения: правильно классифицировать входные векторы как положительные или отрицательные данные. Вероятность для этого рассчитывается как

$$p(positive) = \sigma\left(\sum_j y_j^2 - \theta\right)$$

** goodness function refers to the quality of a model's predictions or its ability to generalize well to new, unseen data. It is often measured by a metric such as accuracy, precision, recall, or F1-score.*

Обучение ногослойных представлений простой функцией

Что мы хотим: обучить скрытые слои и уметь правильно различать наши данные без лишних затрат.

Что делаем: берём FF и нормализуем тем самым длину вектора до отправки в новый скрытый слой.

Что получаем: посчитали goodness, использую нормализованную длину, дальше передали направление вектора.

Ура, скрытый слой использует относительную активность нейронов !

Некоторые эксперименты с алгоритмом

Disclaimer: цель всего этого - показать FF propagation и показать, как она работает на сравнительно небольших нейросетях с несколькими миллионами связей.

Baseline

Немного вводных:

- MNIST датасет написанных от руки цифр
- 50К/10К train/validation во время поиска подходящих гиперпараметров
- тест 10К для подсчета ошибки

Такой выбор датасета обусловлен его изученностью и есть немало результатов обучения небольших нейросетей на нём.

Baseline

Немного вводных:

- MNIST датасет написанных от руки цифр
- 50К/10К train/validation во время поиска подходящих гиперпараметров
- тест 10К для подсчета ошибки

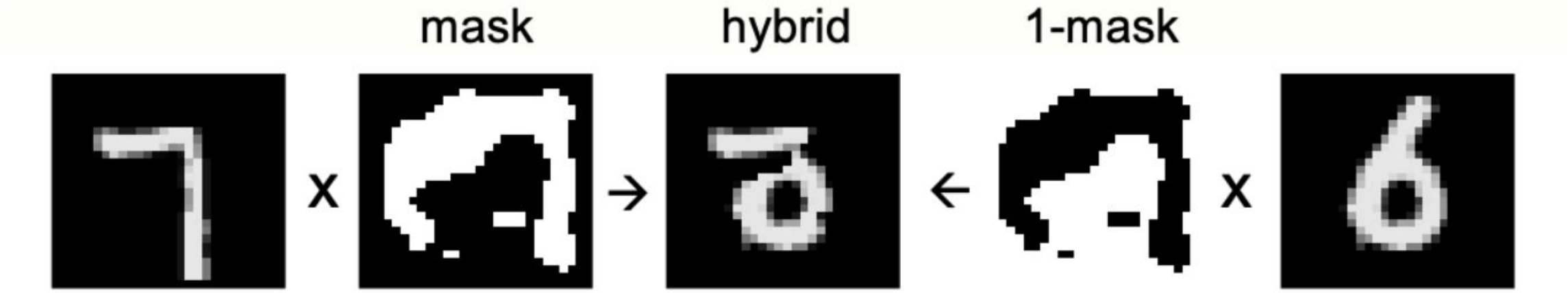
Такой выбор датасета обусловлен его изученностью и есть немало результатов обучения небольших нейросетей на нём.

Результаты: свёрточная нейросеть - 0,6% ошибки;

permutation invariant случай + FNN - 1,4%

permutation invariant случай + FNN + dropout - 1,1%

Unsupervised FF



Unsupervised FF

Только два вопроса меня волнуют:

1. Если у нас есть хороший источник отрицательных данных, обучает ли она эффективные многослойные представления, которые улавливают структуру данных?

2. Откуда берутся отрицательные данные?

Unsupervised FF

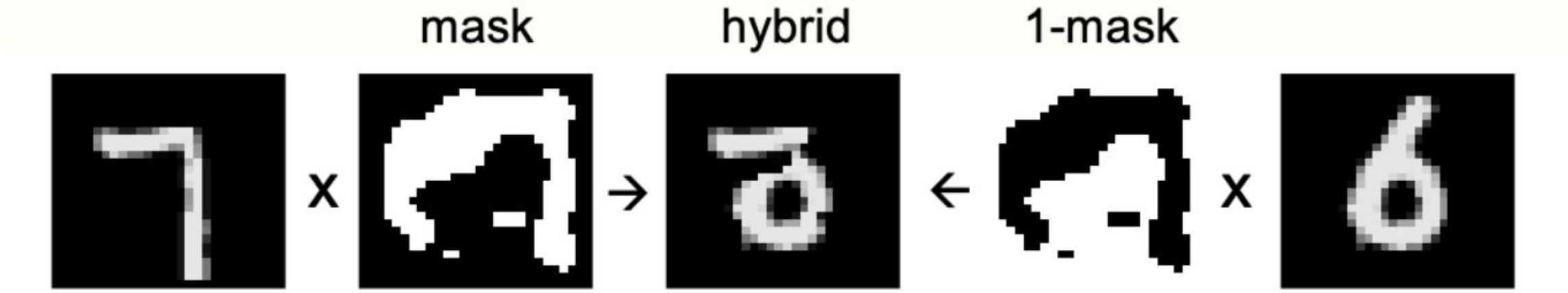
- используем contrastive learning
- положительный пример vs отрицательный пример
- убираем supervised шаг

Unsupervised FF

- используем contrastive learning
- положительный пример vs отрицательный пример
- убираем supervised шаг

  => нужно создать негативные данные с очень разными long range correlations и очень похожими short range correlations

Unsupervised FF



Unsupervised FF

1. 4 скрытых слоя по 2000 relu, обучали 100 эпох. Ошибка 1.37
2. Локальные receptive fields и получить на 60 эпохах. Ошибка 1.16

Описание архитектуры: 1й скрытый слой - 4*4 grid, stride 6, receptieve field 10*10, 128 channels. 2й слой - 3*3 grid, 220 channels в каждой точке, receptieve field - все каналы в квадрате из 4 смежных точек сетки нижнего слоя. 3й слой - grid 2*2, 512 channels.

2000 hidden units per layer.

Supervised FF

Идея: взять лейблы и добавить их в входные параметры.

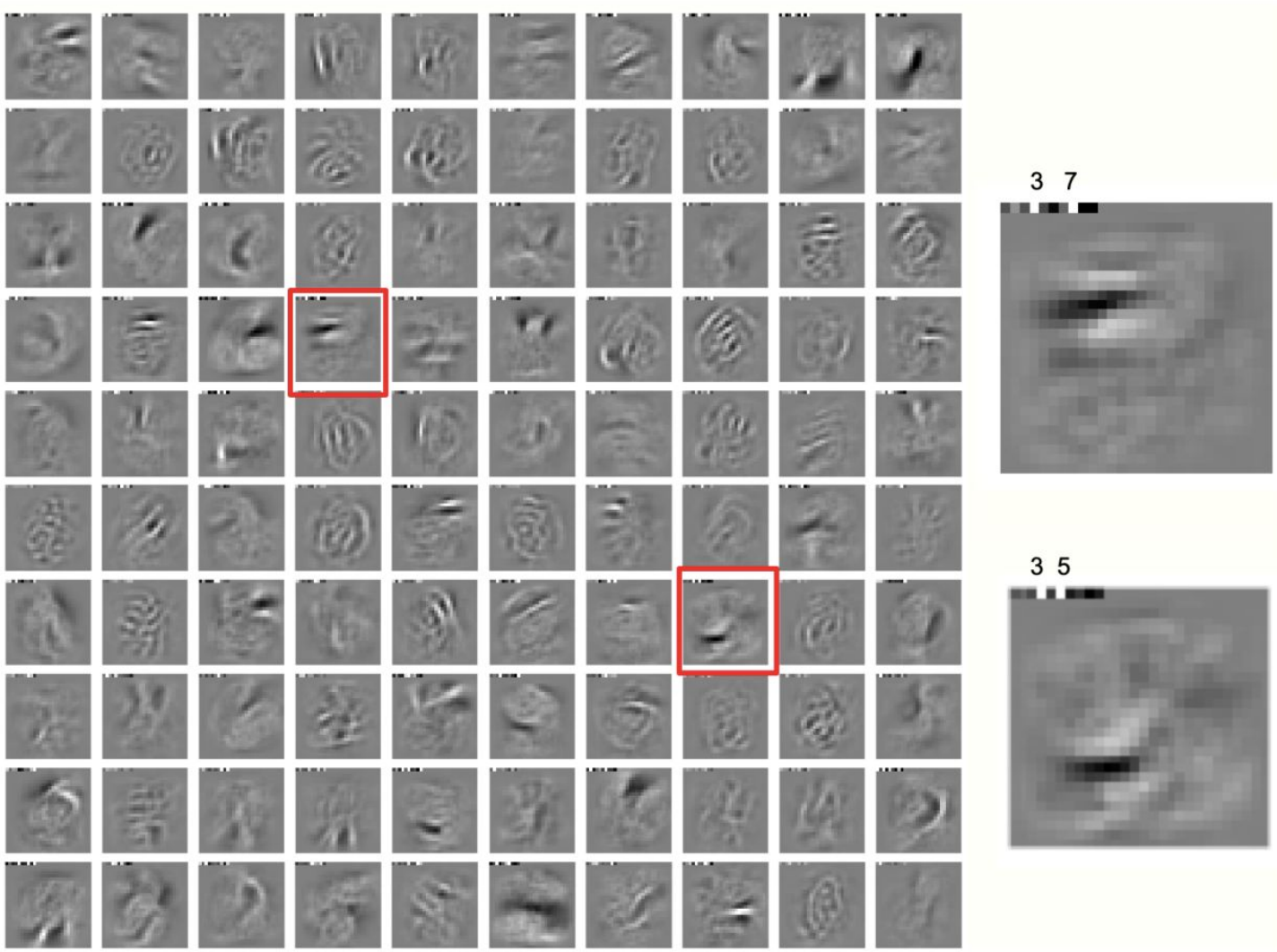
Позитивный пример тут - картинка с верным лейблом, а негативный - картинка с неправильным лейблом

Результаты: полносвязная сеть с 4 слоями с 2000 ReLU **1.36%** ошибки на 60 эпохах.

(Сравнимо с BP на 20 эпохах)

Важно научиться классифицировать цифры моделью!

Top-down эффект в окрестности



CIFAR-10

Вводные: 50000 трейн, 32*32 трехцветные каналы для каждого пикселя. 3072 измерения для картинки.

Проблема - сложный фон картинки, так что полносвязные сети тут в беде.

Сеть: 2-3 скрытых слоя с 3072 ReLU, в скрытом слое топографическая карта 32*32 с 3 скрытыми блоками, рецептивное поле 11 на 11 (на краях карты обрезается).

2 слоя

learning procedure	testing procedure	number of hidden layers	training % error rate	test % error rate
BP		2	0	37
FF min ssq	compute goodness for every label	2	20	41
FF min ssq	one-pass softmax	2	31	45
FF max ssq	compute goodness for every label	2	25	44
FF max ssq	one-pass softmax	2	33	46

3 слоя

learning procedure	testing procedure	number of hidden layers	training % error rate	test % error rate
BP		3	2	39
FF min ssq	compute goodness for every label	3	24	41
FF min ssq	one-pass softmax	3	32	44
FF max ssq	compute goodness for every label	3	21	44
FF max ssq	one-pass softmax	3	31	46

Скорость обучения

Если слои полносвязные, то обновления весов не влияют на нормированный по слою выход самого слоя.

Вектор приращений входящих весов для скрытого нейрона j это

$$\Delta \mathbf{w_j} = 2\epsilon \frac{\partial log(p)}{\partial \sum_j y_j^2} y_j \mathbf{x}$$

yi - активность reLU до нормализации, wj - вектор взодящих весов нейрона, e - скорость обучения.

Скорость обучения

Всё это означает, что можно выполнить одновременно онлайн-обновление весов во многих разных слоях.

Мы стремимся к желаемому значению качества модели, тогда предположим, что входной вектор и все скрытые нормированные векторы имеют длину 1.
Тогда скорость обучения вычисляется как:

$$\epsilon = \sqrt{\frac{S^*}{S_L}} - 1$$

S_L - текущая сумма квадратов активности слоя L до нормализации слоя.

Но - поскольку используются мини батчи, то это не используется в FF

Всем спать !

FF сильно основывается идеей на работе мозга и процессов бодрствования и сна в частности.

И разделение фаз обучения лежит как основная проблема FF с точки зрения биологической модели.

Взаимосвязь с машинами Больцмана

Привет из 1980-х - это сеть стохастических бинарных нейронов с парными связями и одинаковыми весами в обоих направления. При работе без входных данных она обновляет нейроны, переводя их в состояние 1(включения) с вероятностью, равной индикатору от других активных нейронов.