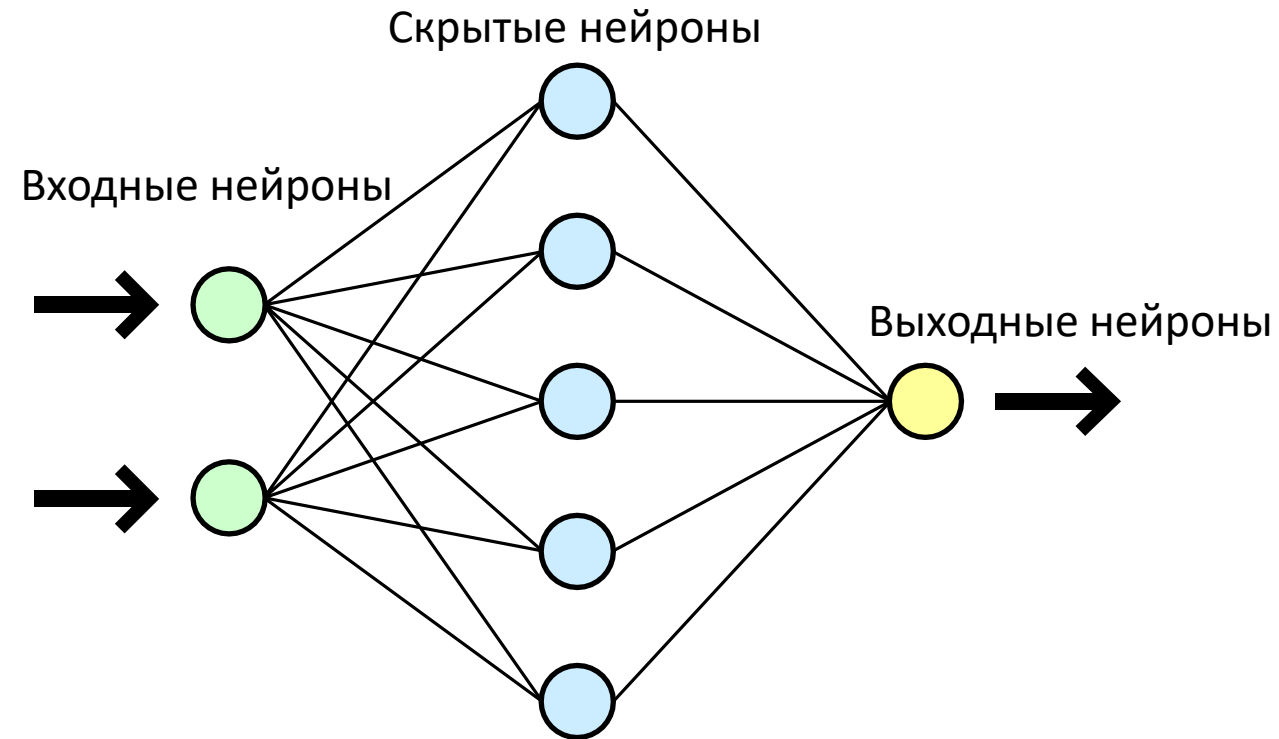


Введение в нейронные сети

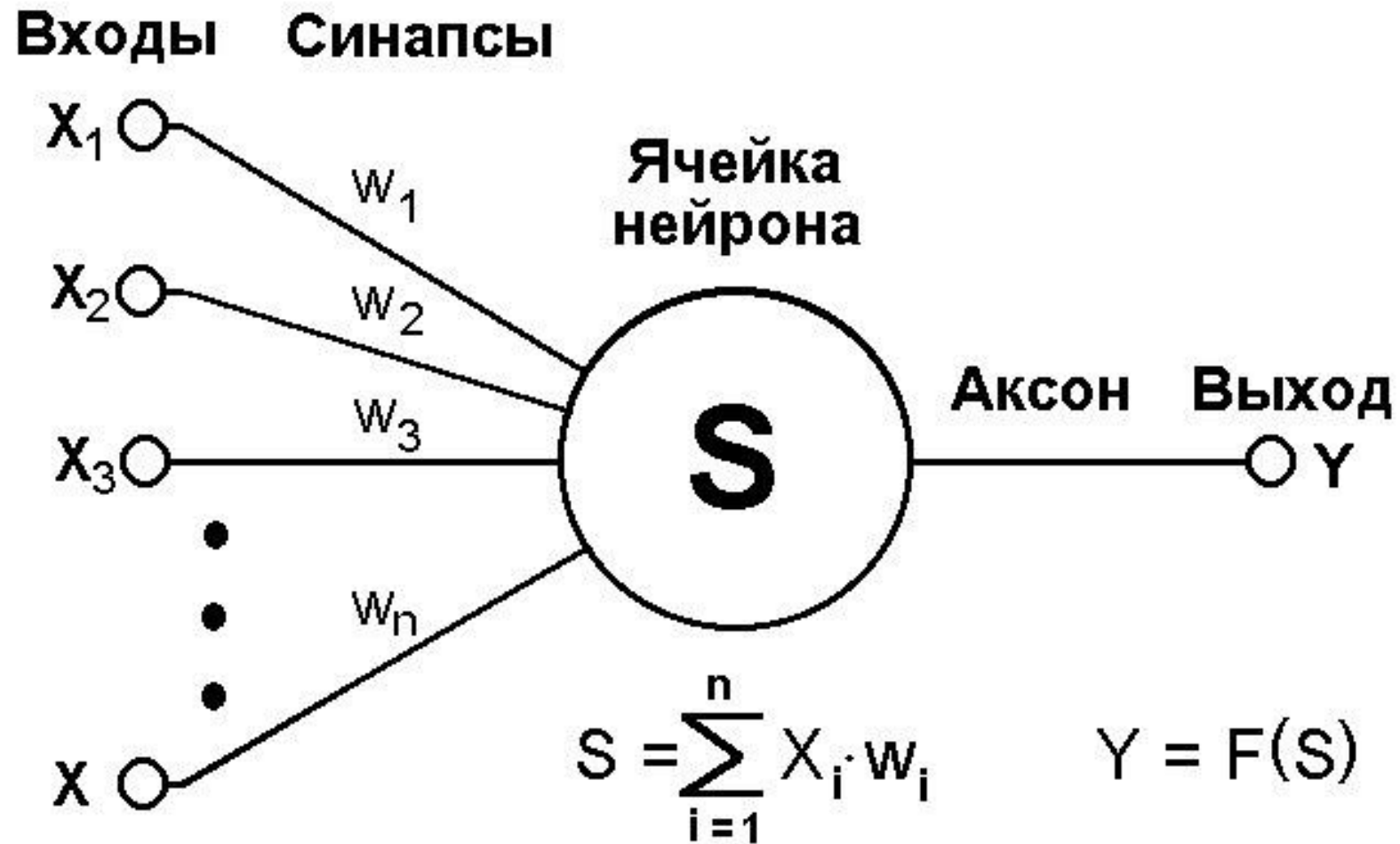
Цыганов Артем, БПМИ-181

Нейронные сети. Определение

- Последовательность нейронов, образующих слои
- Нейроны получают информацию, производят вычисления и передают дальше

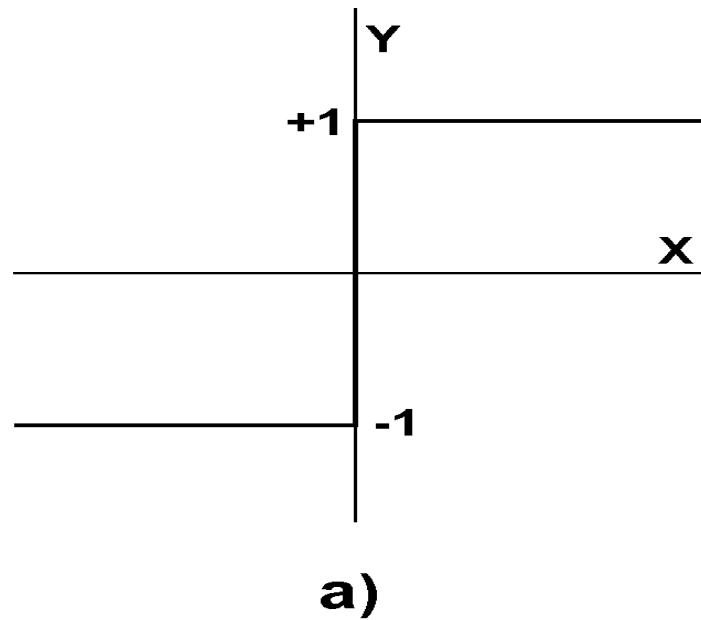


Устройство нейрона

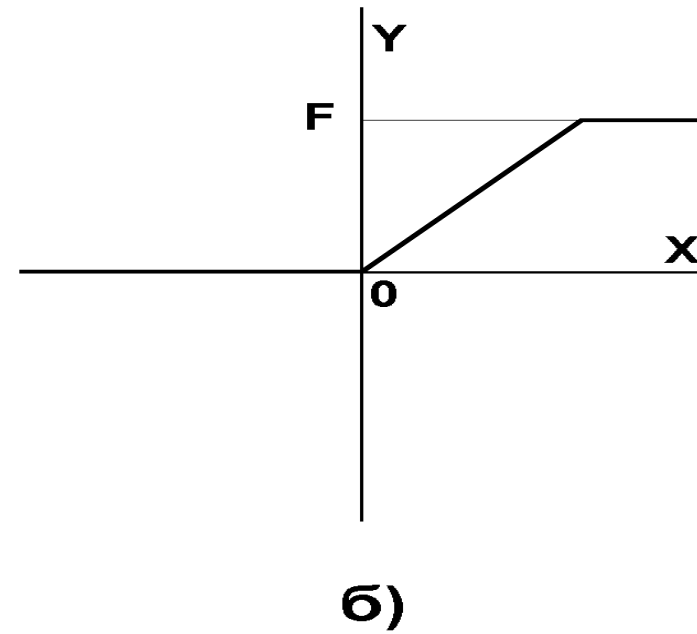


Функции активации

Строгая пороговая

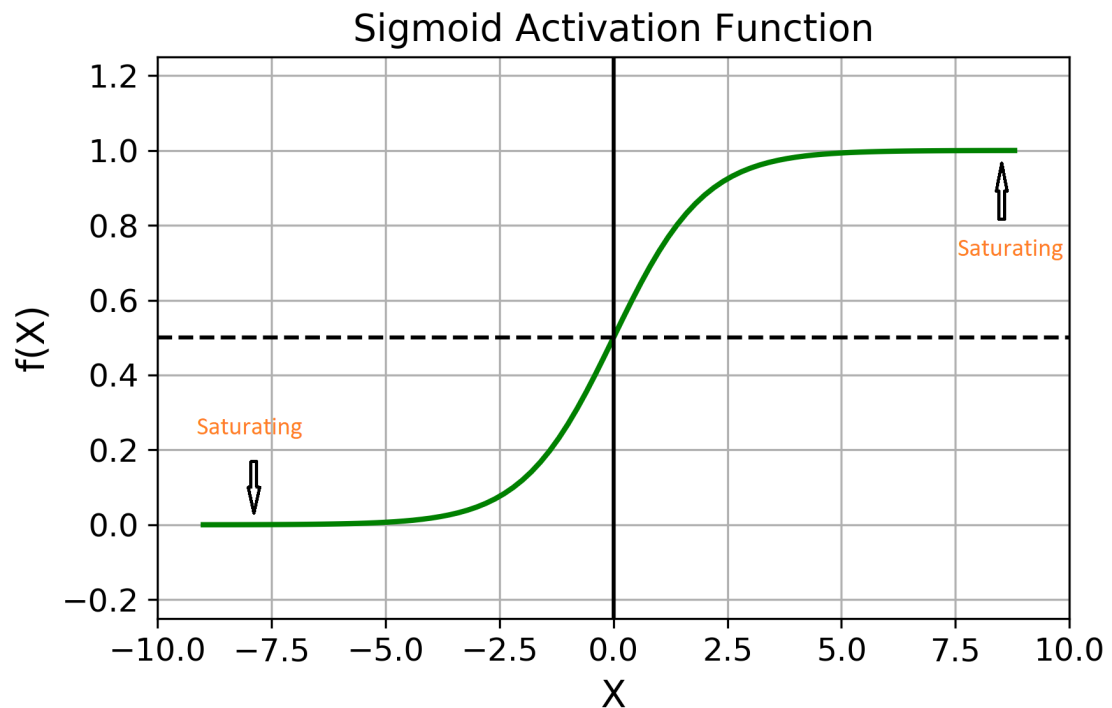


Линейная



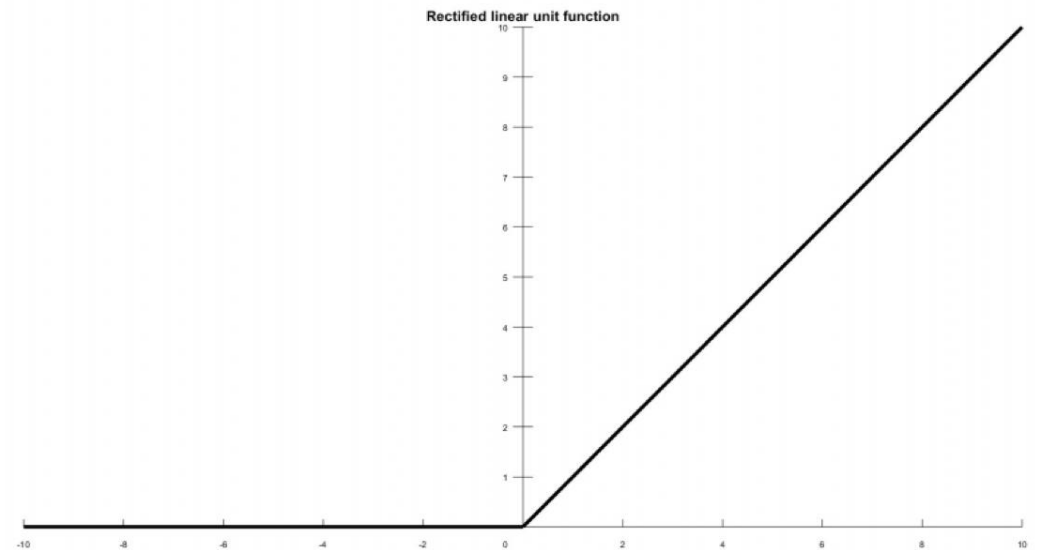
Функции активации

Сигмоидная



ReLu

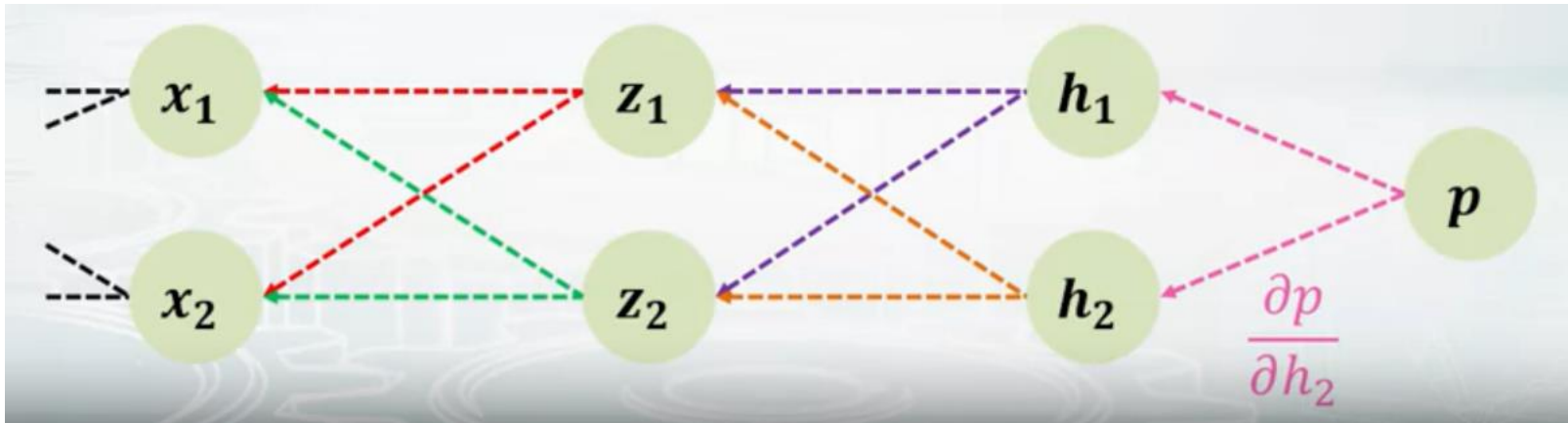
$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$



Применение нейронных сетей

- Распознавание образов и классификация
- Принятие решений и управление
- Кластеризация
- Прогнозирование
- Аппроксимизация
- Сжатие данных и ассоциативная память
- Анализ данных
- Оптимизация

BackPropagation



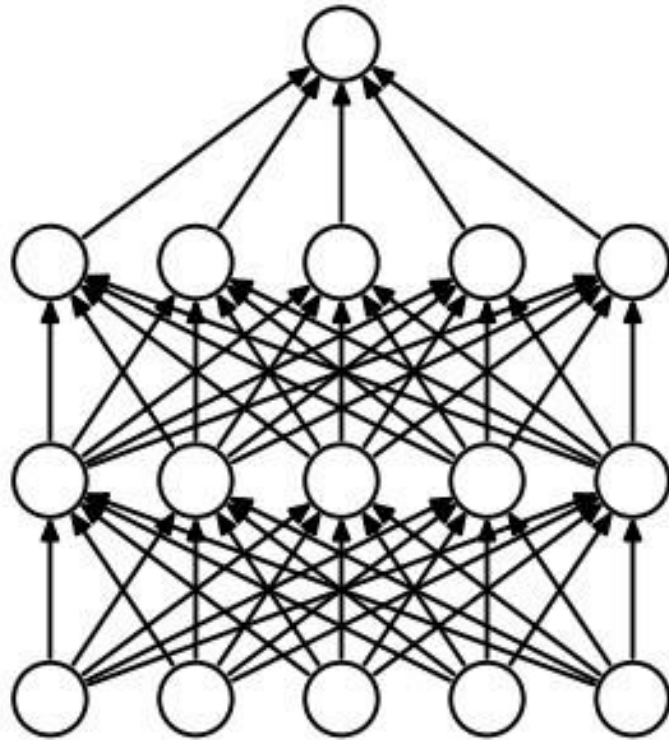
$$h_2 = \sigma(\mathbf{w}_0 + \mathbf{w}_1 z_1 + \mathbf{w}_2 z_2)$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial p} \frac{\partial p}{\partial w_1} = \frac{\partial L}{\partial p} \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial w_1}$$

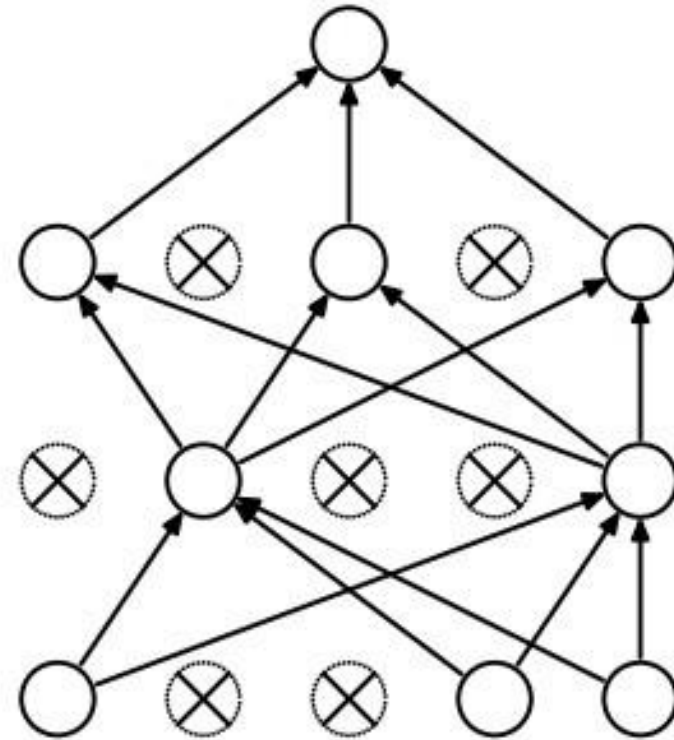
BackPropagation - недостатки

- Паралич сети
- Сходимость к локальному минимуму
- Размер шага

Dropout



(a) Standard Neural Net



(b) After applying dropout.

Прямой dropout

- Применение DropOut к входной проекции вектора

$$f(h) = D \odot a(h)$$

- Тогда применение DropOut к i-ому нейрону

$$O_i = X_i a\left(\sum_{k=1}^{d_i} w_k x_k + b\right) = \begin{cases} a\left(\sum_{k=1}^{d_i} w_k x_k + b\right), & \text{if } X_i = 1 \\ 0, & \text{if } X_i = 0 \end{cases}$$

На этапе обучения: $O_i = X_i a\left(\sum_{k=1}^{d_i} w_k x_k + b\right),$

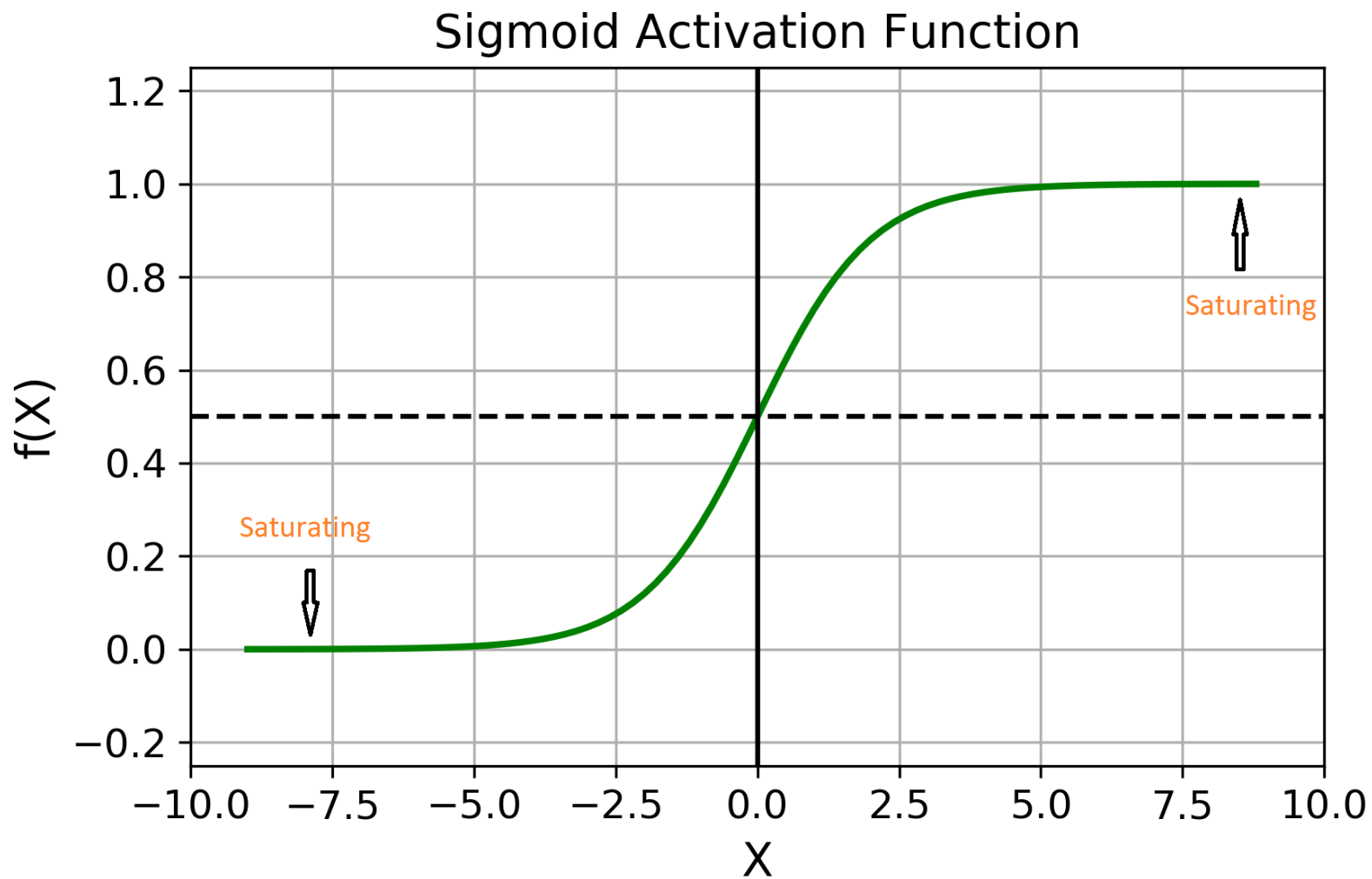
На этапе тестирования: $O_i = qa\left(\sum_{k=1}^{d_i} w_k x_k + b\right)$

Обратный dropout

На этапе обучения: $O_i = \frac{1}{q} X_i a(\sum_{k=1}^{d_i} w_k x_k + b),$

На этапе тестирования: $O_i = a(\sum_{k=1}^{d_i} w_k x_k + b)$

Инициализация сети



Метод Хавьера

- Рассмотрим, что происходит с дисперсией выходных значений линейного нейрона:

$$\text{Var}\left(\sum_{i=1}^{n_{in}} w_i x_i\right) = \sum_{i=1}^{n_{in}} \text{Var}(w_i x_i) = \sum_{i=1}^{n_{in}} \text{Var}(W) \text{Var}(X) = n_{in} \text{Var}(W) \text{Var}(X)$$

- Чтобы сохранить дисперсию входных данных после прохождения через слой, надо, чтобы $\text{Var}(W) = \frac{1}{n_{in}}$
- Аналогичная логика для out
- Тогда в качестве ответа берем их среднее $\text{Var}(W) = \frac{2}{n_{in} + n_{out}}$

Batch normalization

- Преобразуем функцию активации:

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

- Так как к исходным данным мы применили операции масштабирования и сдвига, итоговая функция активации будет:

$$y_i = \gamma \hat{x}_i + \beta$$

Ранняя остановка

- Не должно быть информации о тестовой выборке
- Гиперпараметры подбираются по валидационной выборке
- Подбор гиперпараметров прекращается, если за заданное количество эпох потери не начинают уменьшаться

ИСТОЧНИКИ

- <http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html>
- https://en.wikipedia.org/wiki/Artificial_neural_network
- <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>
- <https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>
- <https://neerc.ifmo.ru/wiki/index.php?title=Batch-normalization>
- <https://habr.com/ru/company/wunderfund/blog/315476/>
- <http://data4.ru/xavier>
- <https://medium.com/zero-equals-false/early-stopping-to-avoid-overfitting-in-neural-network-keras-b68c96ed05d9>
- <https://habr.com/ru/company/wunderfund/blog/330814/>