

Intro to DL, MLP, backpropagation, main terms

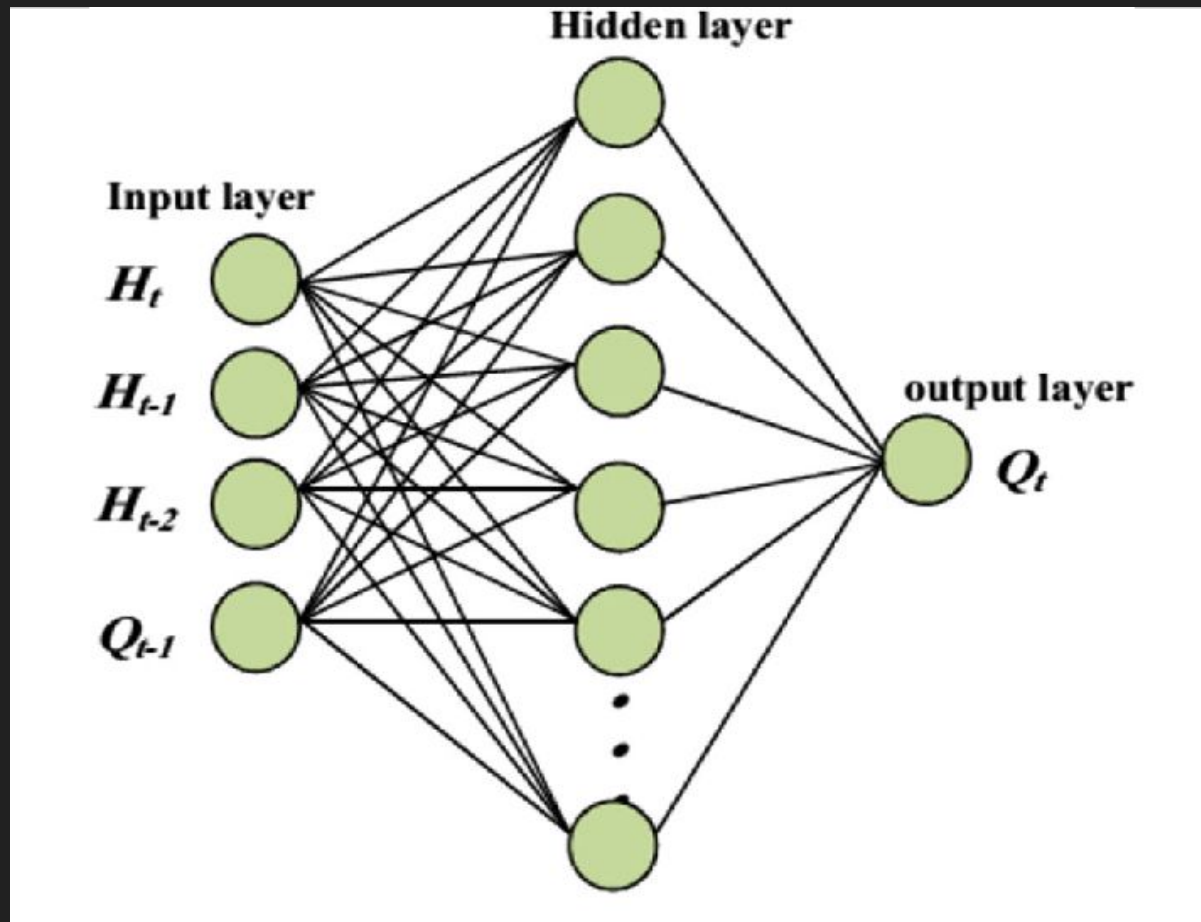
Зачем нужно глубокое (глубинное) обучение?

- Во-первых, это стремление к переходу от построения сложных пайплайнов, каждая компонента которых тренируется сама по себе решать кусочек задачи, к end-to-end обучению всей системы, как одного целого. Так, мы можем обучать не каждый слой отдельно, а все вместе, и не учить какие-нибудь линейные модели поверх случайных лесов, а работать с одной цельной моделью
- Во-вторых, это обучение представлений объектов — информативных признаков описаний, учитывающих структуру данных, построенных лишь на основе самих данных, зачастую неразмеченных. Это позволяет автоматизировать процесс отбора признаков: теперь вместо того, чтобы руками экспертов искать «более информативное» или «более простое» признаковое описание наших объектов, мы можем получить их автоматически, притом используя не только данные, доступные нам для задачи, но и, к примеру, все тексты мира.

Какие виды обучения бывают?

- С учителем (supervised) - имеем размеченные данные
- Без учителя - размеченных данных нет, модель итеративно пытается понять лучшую стратегию (политику) поведения

MLP



Из чего состоит MLP?

Слои:

- **Входной слой** - принимает входные данные
- **Скрытый слой** - обрабатывают данные, выделяя признаки.
- **Выходной слой** - генерирует итоговый результат.

Нейрон:

- Линейная комбинация входов:

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

- Нелинейность (функция активации):

$a = \sigma(z)$ Примеры функций активации:

- Sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$
- ReLU: $\text{ReLU}(z) = \max(0, z)$

Вопросы на подумать?

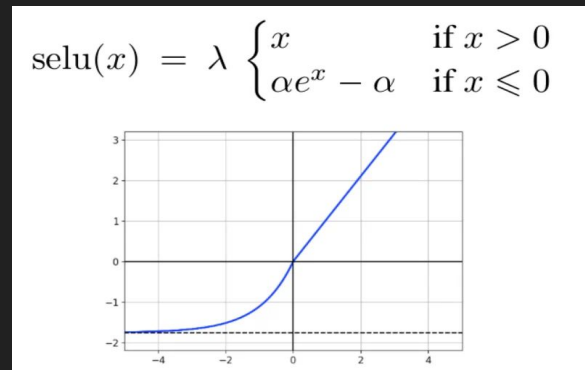
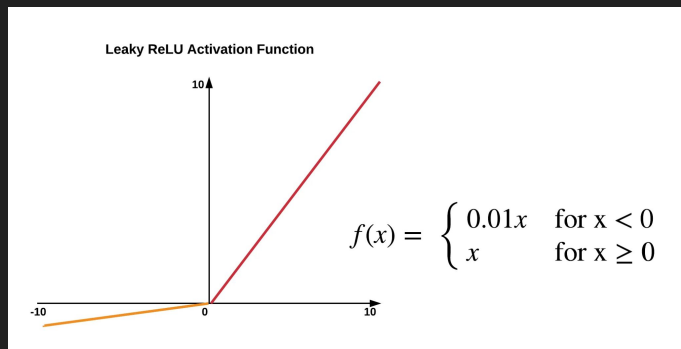
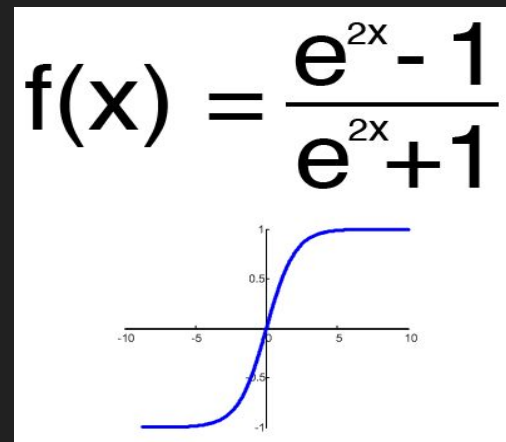
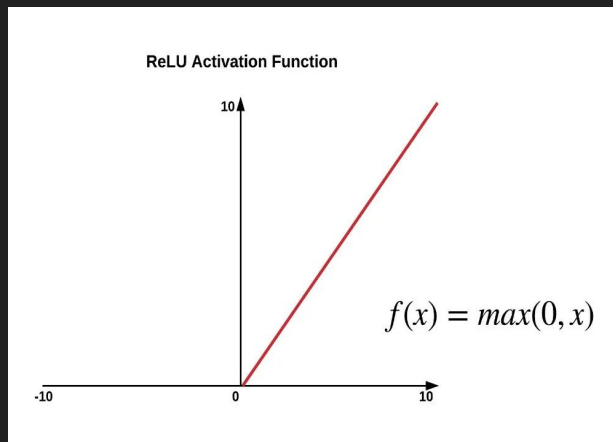
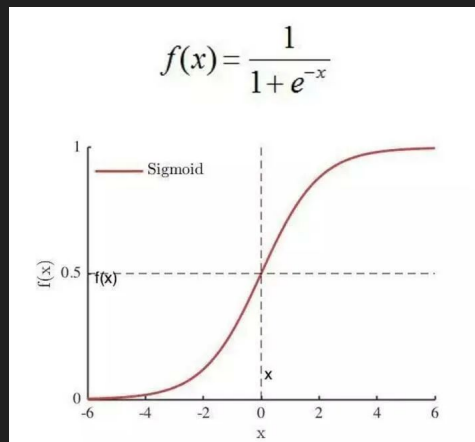
- Чем определяется глубина сети?
- Почему MLP == полносвязный слой (FC) == полносвязная нейронная сеть?
- Зачем нужна функция активации?
- Можно ли без ф-ции активации, что будет если её выкинуть?

Функция активации

Функция активации - нелинейная функция, которая необходима для усложнения нашей сети

Иначе она была бы линейной комбинацией линейных функций => линейной функцией (как с бустингом, крч)

Примеры функций активации



Feed forward

“Прямой проход” нужен, чтобы получить предсказания от сети - ВСЁ



Backpropagation

Здесь начинается самое интересное

С помощью функции потерь мы пытаемся корректировать веса нашей модели

Шаг 1: Вычисление ошибки на выходе сети (например, разница между предсказанием и истинным значением).

Шаг 2: Обратное распространение ошибки через все слои сети.

Шаг 3: Применение правила цепочки для вычисления градиентов весов и смещений каждого слоя сети.

Шаг 4: Обновление весов с использованием градиентного спуска

Chain rule

Если у нас есть составная функция $y = f(g(x))$, то производная этой функции по x может быть найдена с использованием правила цепочки:

$$\frac{dy}{dx} = \frac{dy}{dg} \cdot \frac{dg}{dx}$$

Здесь:

- $f(g(x))$ — составная функция, где $g(x)$ — внутренняя функция, а f — внешняя.
- $\frac{dy}{dg}$ — производная внешней функции f по g .
- $\frac{dg}{dx}$ — производная внутренней функции g по x .

То есть: мы можем разбить сложную функцию на несколько и спокойно посчитать частные производные

Пример:

Предположим, у нас есть две функции:

- $z = g(x) = x^2$
- $y = f(z) = \sin(z)$

Чтобы найти производную $\frac{dy}{dx}$, мы применяем правило цепочки:

$$\frac{dy}{dx} = \frac{dy}{dz} \cdot \frac{dz}{dx}$$

1. Сначала находим производную $f(z)$ по z : $\frac{dy}{dz} = \cos(z)$.
2. Затем находим производную $g(x)$ по x : $\frac{dz}{dx} = 2x$.
3. Умножаем эти производные:

$$\frac{dy}{dx} = \cos(z) \cdot 2x = \cos(x^2) \cdot 2x$$

На подумать

- Как инициализируются веса для первого feed forward?
- Почему большие веса - это плохо?
- Зачем нам нужна loss функция, если мы смотрим на метрики? Какая особенность у этой функции должна быть?
- Почему нейросеть нельзя сделать ооочень глубокой? (раньше даже 18 слоев было много)

Гиперпараметры

- Задаются нами, не обучаемы
- Влияют на сходимость, качество и устойчивость модели
- Контролируют процесс обучения

Параметры

- Обучаемы, зависят от данных
- Корректируются во время обучения
- Определяют качество модел

ResNet

