

1. Что такое слой активации в сверточных нейронных сетях?

Слой активации в сверточных нейронных сетях – это слой, на котором значения входного тензора преобразуются в выходные значения с помощью некоторой функции (функции активации).

После полносвязного (и, обычно, после любого другого обучаемого) слоя всегда ставится слой активации.

2. Для чего нужна регуляризация весов?

Регулирование качества информации или добавление ограничений на информацию, которую модели будет позволено сохранить. Если сеть может позволить себе сохранить только небольшое количество шаблонов, процесс оптимизации заставит ее сосредоточиться на наиболее существенных из них, что увеличит шансы на достижение более высокого уровня общности.

Борьба с переобучением таким способом называется регуляризацией.

То есть регуляризация в машинном обучении – метод добавления некоторых дополнительных ограничений к условию с целью решить некорректно поставленную задачу или предотвратить переобучение.

3. Чем чревато использования большой скорости обучения?

Большая скорость обучения может препятствовать сходимости, и как следствие функция ошибок будет колебаться вокруг глобального минимума и не достигнет его.

Таким образом, более высокая скорость обучения помогает нейросети делать большие шаги, т.е. большая скорость сходимости в направлении минимальной ошибки. Однако, когда сеть приближается к нижней части кривой ошибки, эти длинные шаги могут препятствовать сходимости (подобно тому, как человеку, делающему длинные шаги, может быть трудно попасть прямо в середину небольшого круга, нарисованного на полу).

4. Что такое переобучение модели?

Переобучение (overfitting) происходит тогда, когда модели машинного обучения показывают худшую точность на новом наборе данных по сравнению с тренировочным.

Переобучение – это результат чрезмерной подгонки параметров модели к зависимостям, содержащимся в обучающем множестве. Если происходит переобучение, то модель не приобретает способности к обобщению – возможности распространять обнаруженные на обучающем множестве зависимости и закономерности на новые данные. Такая модель на практике оказывается бесполезной.

5. В чем особенность оптимизатора rmsprop?

Решает проблему стремительного уменьшения скорости обучения. RMSProp или Root Mean Square Propagation автоматически уменьшит размер шагов градиента к минимумам, когда шаги слишком велики (большие шаги делают нас склонными к выбросам). Оптимизатор учитывает историю обновлений весов, часто обновляемые обновляются меньше.

Сумма квадратов градиентов в данном алгоритме заменяется на экспоненциально затухающее среднее всех предыдущих квадратов градиентов, то есть в большей степени учитываются последние значения частных производных.

$$E[g^2]_N = \gamma E[g^2]_{N-1} + (1 - \gamma) g_N^2$$

Тогда правило обновления весовых коэффициентов принимает вид:

$$w^{N+1} = w^N - \frac{\eta}{\sqrt{E[g^2]_N + \varepsilon}} g_N$$

Так как знаменатель представляет собой корень из среднего квадратов градиентов (root mean square), обозначим знаменатель в формуле следующим образом:

$$w^{N+1} = w^N - \frac{\eta}{RMS[g]_N} g_N$$

Последнее уравнение является шагом обновления весов. Коэффициент сохранения η обычно выбирается равным 0,9.

6. Для чего добавлена метрика mae?

Для регрессии используются иные метрики оценки, нежели при классификации; понятие точности неприменимо для регрессии, поэтому для оценки качества часто применяется средняя абсолютная ошибка между фактическим и прогнозируемым значениями (Mean Absolute Error, MAE).

MAE устойчива к выбросам, в отличие от среднего арифметического, т.е. большое отклонение данных не повлияет на общую ошибку.

7. В чем недостаток функции relu?

ReLU не всегда достаточно надежны и в процессе обучения могут выходить из строя («умирать» – Dying ReLU problem). Например, большой градиент, проходящий через ReLU, может привести к такому обновлению весов, что данный нейрон никогда больше не активируется. Если это произойдет, то, начиная с данного момента, градиент, проходящий через этот нейрон, всегда будет равен нулю. Соответственно, данный нейрон будет необратимо выведен из строя. Например, при слишком большой скорости обучения (learning rate), может оказаться, что до 40% ReLU «мертвы» (то есть, никогда не активируются). Эта проблема решается посредством выбора надлежащей скорости обучения.