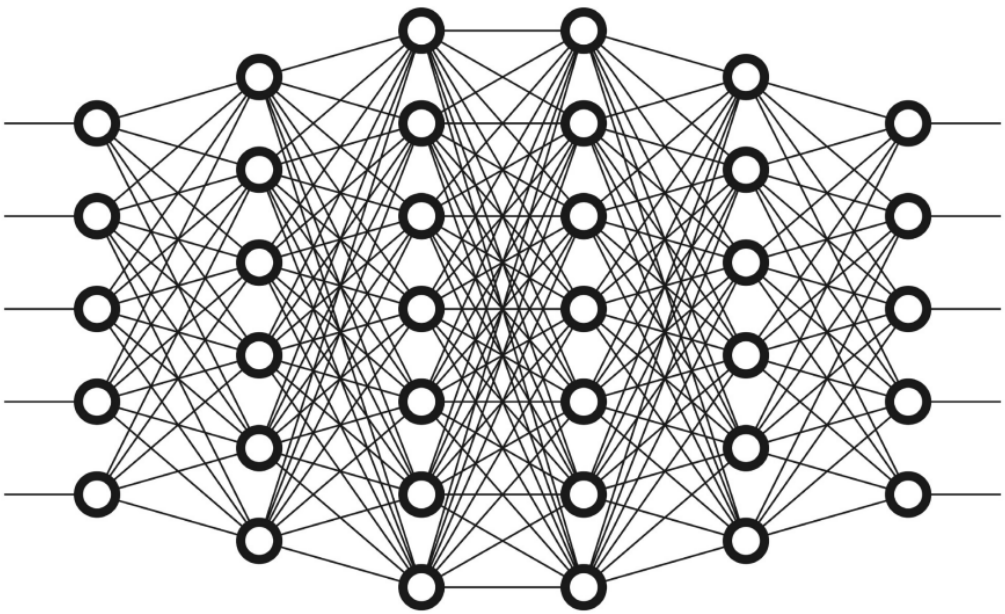
Особенности настройки глубоких нейросетей

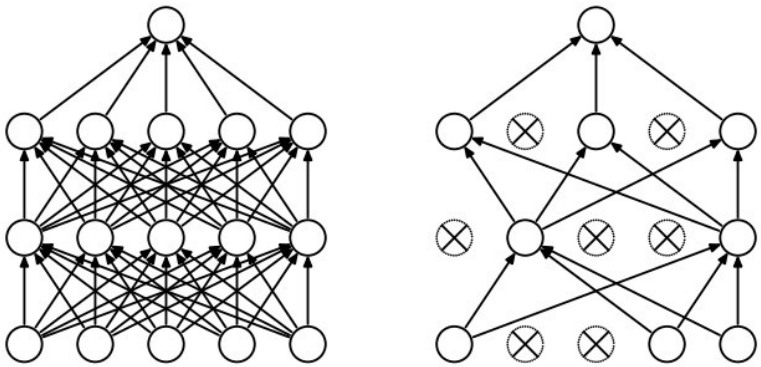
Виктор Китов

v.v.kitov@yandex.ru



Содержание

DropOut: обучение



(a) Standard Neural Net (b) After applying dropout.

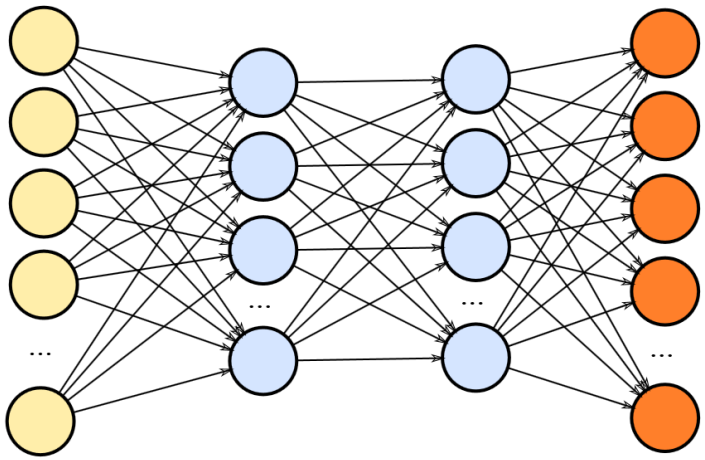
- Для каждого минибатча каждый нейрон, кроме выходных и входных отбрасывается с вероятностью $(1-p)$ и оставляется с вероятностью p.
  - независимо для каждого нейрона
- Это уменьшает переобучение, препятствуя со-настройке нейронов
- По сути, мы учим ансамбль прореженных моделей.
- В среднем нейрон получает на вход $\sum_i (pw_i x_i + (1-p) w_i 0) = \sum_i pw_i x_i$.

DropOut: применение

- Применение модели: оставляются все нейроны.
- Для компенсации изменений, выход каждого нейрона домножается на p.
  - для уменьшения вычислений можно:
    - обучение: выход x_i/p, когда x_i оставлен.
    - прогноз: выход x_i без изменений.

Содержание

Перенормировка активаций: мотивация



- SGD $w := w - \varepsilon \nabla_w \mathcal{L}(x, y)$ обновляет все веса на всех слоях одновременно.
- Распределение выходов меняется, и поздние слои должны обучатся снова.
- Также вход может сдвинуться в область малых градиентов нелинейности.
- Перенормировка активаций (batch normalization) частично это решает.

BatchNorm: идея

- Нормализуем выходы на промежуточных слоях:

$$\tilde{x}_k = \frac{x_k - \mu_k}{\sigma_k}, \quad \mu_k = \mathbb{E}x_k, \sigma_k = \sqrt{Var(x_k)}$$

  - гарантируем $\mathbb{E}\tilde{x}_k = 0,\ \mathrm{Var}\,\tilde{x}_k = 1$ после обновления весов на предыдущих слоях.
    - обучение быстрее для поздних слоёв
- **Обучение:**
  - проблема: не знаем μ_k, σ_k
    - изменяются динамически с обновлением весов
  - решение: оценим по текущему минибатчу (должен быть достаточного размера)
- **Применение:**
  - распределение x_k фиксировано, так что можем оценить μ_k, σ_k по всей обучающей выборке.
  - более эффективно: оценки μ_k, σ_k с последовательности последних минибатчей.

BatchNorm: основной алгоритм

$$\tilde{x}_k = \alpha_k \frac{x_k - \mu_k}{\sqrt{\sigma_k^2 + \varepsilon}} + \beta_k, \quad \mu_k = \bar{x}_k,\ \sigma_k = \sqrt{Var(x_k)},\ \varepsilon = 10^{-6}.$$

- **Обучение:**
  - μ_k, σ_k по минибатчу
  - α_k, β_k выходное стд. отклонение и среднее.
    - обучаются в процессе настройки сети
  - мотивация:
    - можем отменить нормализацию (например в задаче предсказания времени суток по фото)
    - возможность лучше подстроиться под нелинейность (не обнулять вход в половине случаев для ReLU)
- **Применение:**
  - μ_k, σ_k оценены по широкому классу объектов.
  - α_k, β_k фиксированы.