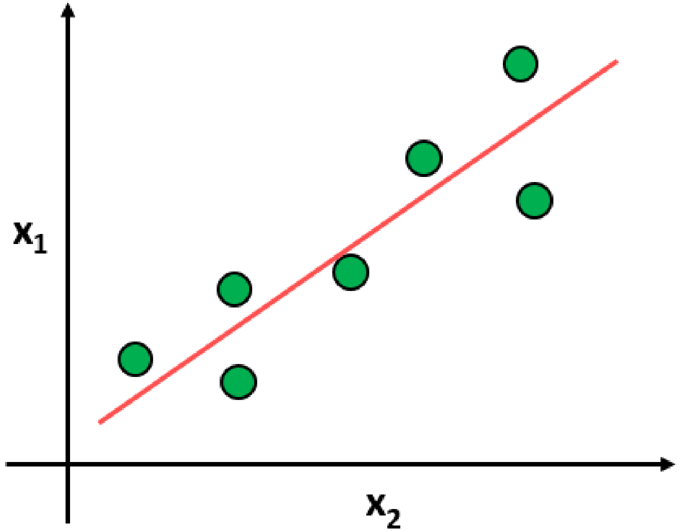
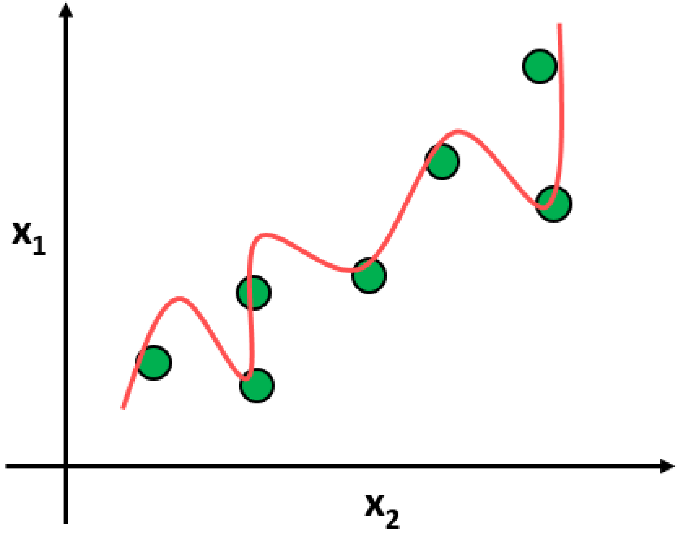
Regularization





Regularization

- Neural networks typically have thousands, if not millions of parameters
  - Usually, the dataset size smaller than the number of parameters
- Overfitting is a grave danger
- Regularization is crucial to avoid overfitting
- Possible regularization methods
  - ℓ_2-regularization
  - ℓ_1-regularization
  - Dropout
  - …

ℓ_2-regularization

- Most important (or most popular) regularization

$$w^* \leftarrow \arg\min_w \sum_{(x,y)\subseteq(X,Y)} \mathcal{L}(y, a_L(x; w_{1,\dots,L})) + \frac{\lambda}{2}\sum_l \|w_l\|^2$$

- The ℓ_2-regularization is added to the gradient descent update rule

$$w_{t+1} = w_t - \eta_t(\nabla_\theta \mathcal{L} + \lambda w_l) \Longrightarrow$$
$$w_{t+1} = (1 - \lambda\eta_t)w^{(t)} - \eta_t \nabla_\theta \mathcal{L}$$

"Weight decay", because weights get smaller

- λ is usually about $10^{-1}, 10^{-2}$

ℓ_1-regularization

- ℓ_1-regularization is one of the most important regularization techniques

$$\mathrm{w}^* \leftarrow \arg\min_w \sum_{(x,y)\subseteq(X,Y)} \mathcal{L}(y, a_L(x; w_{1,\ldots,\mathrm{L}})) + \frac{\lambda}{2} \sum_l |w_l|$$

- Also ℓ_1-regularization is added to the gradient descent update rule

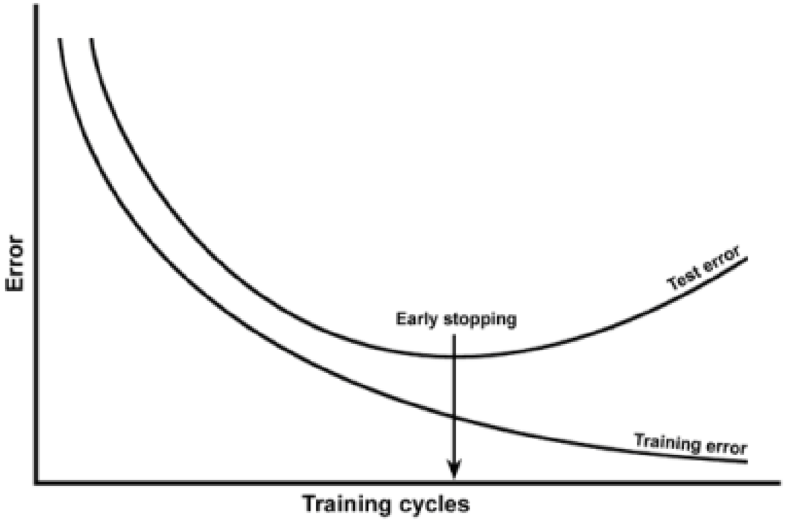
$$w_{t+1} = w_t - \eta_t \left(\nabla_\theta \mathcal{L} + \lambda \frac{w^{(t)}}{|w^{(t)}|} \right)$$

Sign function

- ℓ_1-regularization → sparse weights
  - $\lambda \nearrow$ → more weights become 0

Early stopping

- Monitor performance on a separate validation set
- Training the network will decrease training error, as well validation error Stop when validation error starts increasing
  - This quite likely means the network starts to overfit
- For a linear model equivalent to ℓ_2-regularization (link)



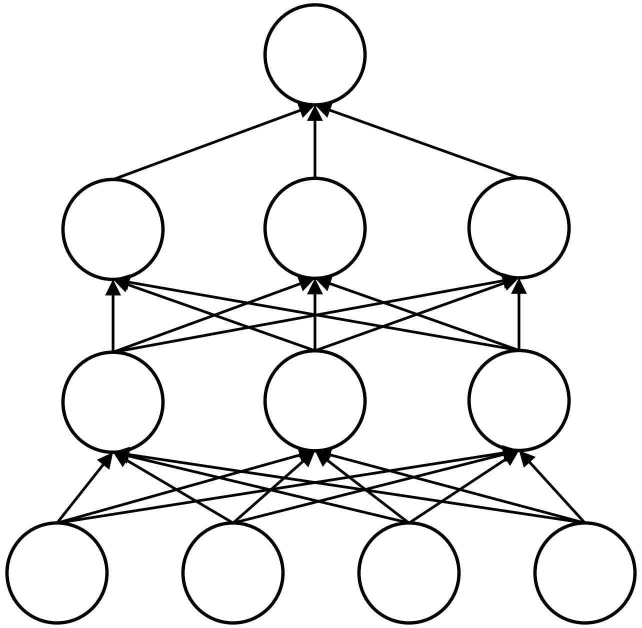
Dropout [Srivastava2014]

- During training randomly set activations to 0
  - Neurons sampled at random from a Bernoulli distribution with $p = 0.5$
- During testing all neurons are used
  - Neuron activations reweighted by p
- Benefits
  - Reduces complex co-adaptations or co-dependencies between neurons
  - Every neuron becomes more robust
  - Decreases overfitting

Dropout

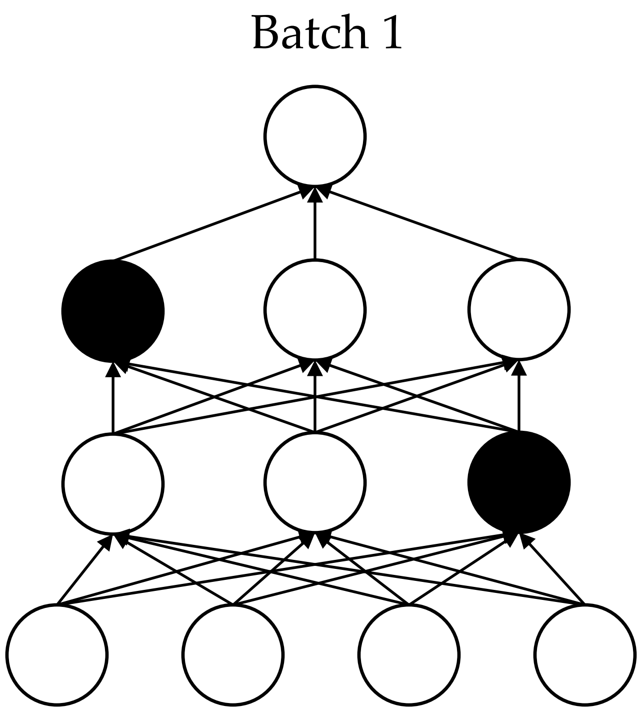
- Effectively, a different architecture for every input batch during training
  - Similar to model ensembles

Original model



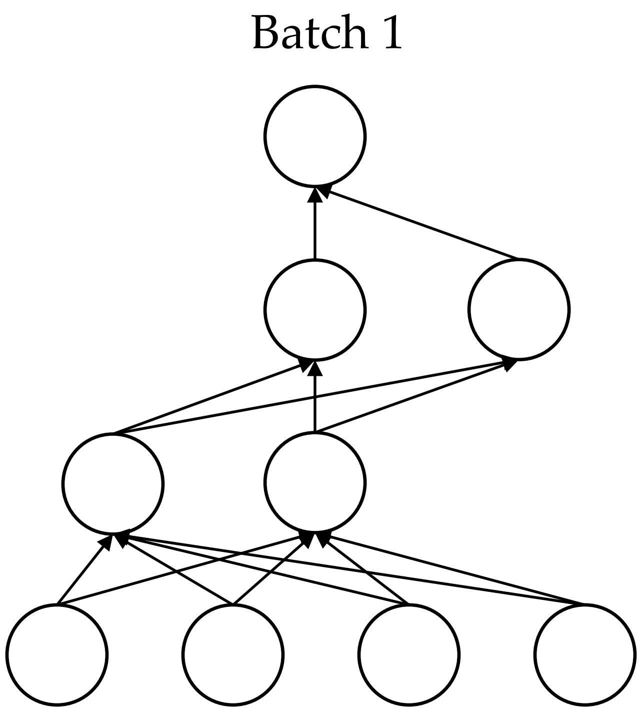
Dropout

- Effectively, a different architecture for every input batch during training
  - Similar to model ensembles



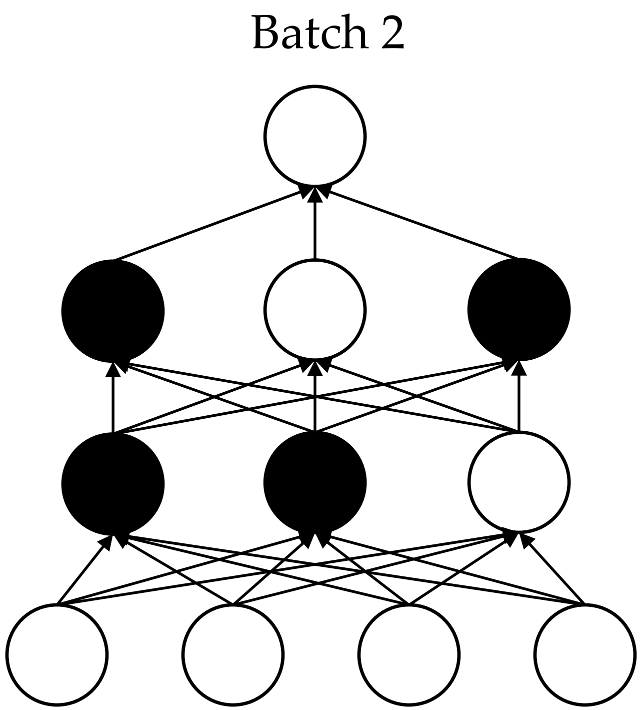
Dropout

- Effectively, a different architecture for every input batch during training
  - Similar to model ensembles



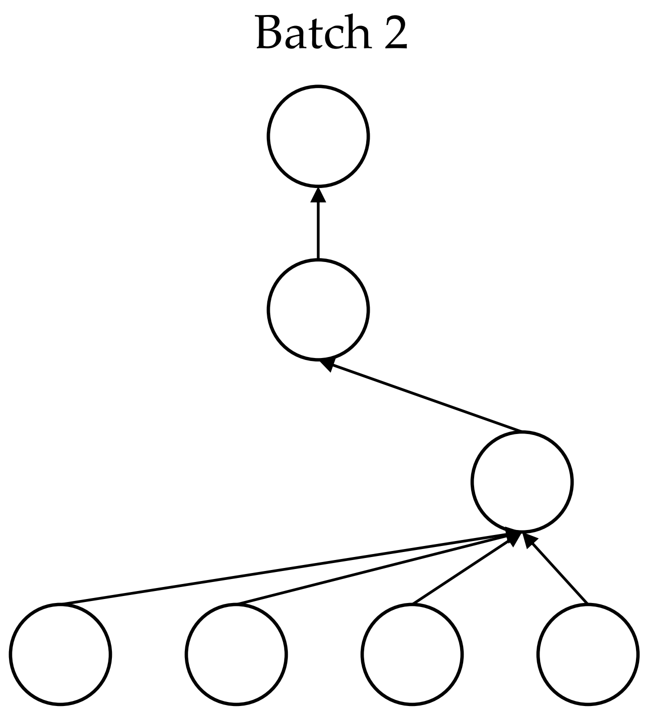
Dropout

- Effectively, a different architecture for every input batch during training
  - Similar to model ensembles



Dropout

- Effectively, a different architecture for every input batch during training
  - Similar to model ensembles



Data augmentation

Original

Flip

Random crop

Contrast

Tint