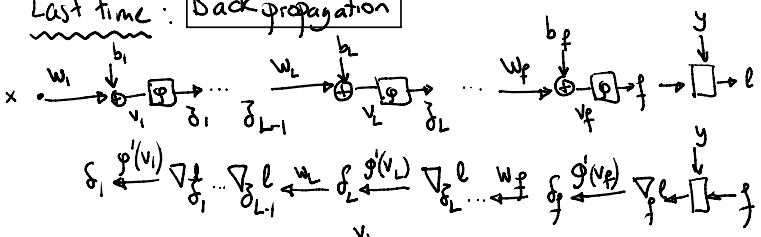
ECE/CS 559 Lecture 11 T 10/1

Last time: **Backpropagation**



Forward: $\ell(z_L) = \ell\big(\varphi\big(\underbrace{W_L z_{L-1} + b_L}_{v_L}\big)\big)$

Backward: $\nabla_{z_{L-1}} \ell = W_L^T \big(\varphi'(v_L) \circ \nabla_{z_L} \ell\big) = W_L^T \delta_L$ (element wise)

Gradients: $\nabla_{W_L} \ell = \big(\varphi'(v_L) \circ \nabla_{z_L} \ell\big) z^T = \delta_L z^T$, $\nabla_{b_L} \ell = \delta_L$

- When learning neural networks, issue is that we train on the observed contexts x. However, we mostly care about performance on new unobserved contexts!
- Performing well in unobserved contexts = **generalization**
  Overfitting prevents generalization (unless it's mild)
- How do we quantify this?

  Training Data $= \{(x,y), \dots\}$ ↳ use this to build NN

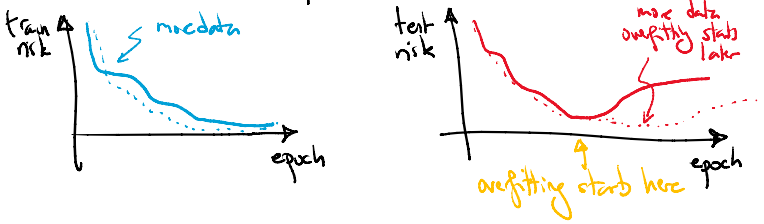
  Testing Data $= \{(x',y'), \dots\}$ ↳ use this to assess generalization
- We must be careful!
  We can't keep on using testing data to refine the NN, otherwise we may overfit it too and not generalize to newer data.

① Overfitting

- Tendency to overexplain the training data.
- Overexplain = when many explanations describe experience equally well, choose one that adds many complicated details and twists (typically what conspiracy theories do)
- Example: polynomial fitting

- Typical property leading to this: high sensitivity to changes in data

- Typical behavior of train vs. test risk:



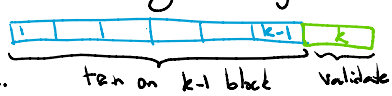
- Observations:
  - Stopping early could help alleviate overfitting
    Why? The NN will not fit the noise.
    How? By staying closer to its initialization (staying "simpler")
  - More training data reduces overfitting, delays its onset.

② Cross-Validation

- Since training risk is not a good enough proxy to testing risk, we could hold out some of the training data for validation:

Training Data $= \{(x,y), \dots\}$ (train on this often) Validation Data $= \{(x'',y''), \dots\}$ (check if okay only sometimes) Testing Data $= \{(x',y'), \dots\}$ (test on this only once)

- What is validation good for?
  - Choose what epoch to stop at.
  - Choose between possible architectures, learning rates, etc. ("hyper parameters")
  - These choices are "coarser" than choosing the weights & biases
- k-Fold Cross-validation:



Change validation block, repeat. Average.

③ Regularization

- Cross-validation helps see if we overfit. Regularization helps prevent it.
- The word comes from inverse problems in physics:
  - To decide between many possible explanations, we need to favor some.
  - We tend to favor "simple" ones (Occam's Razor) (e.g., low energy)
- E.g., For polynomials and NNs, favor small weights:

  $\min_w R(w)$ for $\|w\|^2 \le C$ $\iff$ $\min_w R(w) + \lambda \|w\|_2^2$ (regularized/penalized risk)
- What does it mean? Prefer smoother functions: (sparser)
- Why does it work? Can't overfit noise
  Regularized risk is a better proxy for test-risk
  $\|w\|_1 = |w_1| + \dots + |w_n|$ (other choice)
- How does it affect backpropagation? Simple!

  $\nabla_w \big(R(w) + \lambda \underbrace{\|w\|_2^2}_{\|w\|_2}\big) = \nabla_w R + 2\lambda w$ ($\operatorname{sign}(w)$)

Note: Any change that helps prefer some w over others is regularization. This introduces what we call inductive bias, which (if right) improves generalization.