Neural networks

Training neural networks - regularization
**Topics:** stochastic gradient descent (SGD)

- Algorithm that performs updates after each example
  - initialize $\theta$  
    \[ \theta \equiv \{ W^{(1)}, b^{(1)}, \ldots, W^{(L+1)}, b^{(L+1)} \} \]
  - for N iterations
    - for each training example $(x^{(t)}, y^{(t)})$
      \[ \Delta = -\nabla_\theta l(f(x^{(t)}; \theta), y^{(t)}) - \lambda \nabla_\theta \Omega(\theta) \]
      \[ \theta \leftarrow \theta + \alpha \Delta \]
  - training epoch
  - iteration over all examples

- To apply this algorithm to neural network training, we need
  - the loss function $l(f(x^{(t)}; \theta), y^{(t)})$
  - a procedure to compute the parameter gradients $\nabla_\theta l(f(x^{(t)}; \theta), y^{(t)})$
  - the regularizer $\Omega(\theta)$ (and the gradient $\nabla_\theta \Omega(\theta)$)
  - initialization method
Topics: L2 regularization

\[ \Omega(\theta) = \sum_k \sum_i \sum_j (W_{i,j}^{(k)})^2 = \sum_k \|W^{(k)}\|_F^2 \]

• Gradient: \( \nabla_{W^{(k)}} \Omega(\theta) = 2W^{(k)} \)

• Only applied on weights, not on biases (weight decay)

• Can be interpreted as having a Gaussian prior over the weights
Topics: L1 regularization

\[ \Omega(\theta) = \sum_k \sum_i \sum_j |W_{i,j}^{(k)}| \]

- Gradient: \( \nabla_{W^{(k)}} \Omega(\theta) = \text{sign}(W^{(k)}) \)
  - where \( \text{sign}(W^{(k)})_{i,j} = 1_{W_{i,j}^{(k)}>0} - 1_{W_{i,j}^{(k)}<0} \)
- Also only applied on weights
- Unlike L2, L1 will push certain weights to be exactly 0
- Can be interpreted as having a Laplacian prior over the weights
Topics: bias-variance trade-off

- Variance of trained model: does it vary a lot if the training set changes
- Bias of trained model: is the average model close to the true solution
- Generalization error can be seen as the sum of the (squared) bias and the variance