Neural networks
Training neural networks - parameter initialization
**Topics:** stochastic gradient descent (SGD)

- Algorithm that performs updates after each example
  - initialize $\boldsymbol{\theta}$ \quad ( $\boldsymbol{\theta} \equiv \{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{W}^{(L+1)}, \mathbf{b}^{(L+1)}\}$
  - for N iterations
    - for each training example \ (\mathbf{x}^{(t)}, y^{(t)})
      \[ \begin{align*}
      \Delta &= -\nabla_{\boldsymbol{\theta}} l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)}) - \lambda \nabla_{\boldsymbol{\theta}} \Omega(\boldsymbol{\theta}) \\
      \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} + \alpha \Delta 
      \end{align*} \]
      \[ \text{training epoch} \]
      \[ \text{iteration over all examples} \]
- To apply this algorithm to neural network training, we need
  - the loss function $l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)})$
  - a procedure to compute the parameter gradients $\nabla_{\boldsymbol{\theta}} l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)})$
  - the regularizer $\Omega(\boldsymbol{\theta})$ (and the gradient $\nabla_{\boldsymbol{\theta}} \Omega(\boldsymbol{\theta})$
  - initialization method
Topics: initialization

• For biases
  ‣ initialize all to 0

• For weights
  ‣ Can't initialize weights to 0 with tanh activation
    - we can show that all gradients would then be 0 (saddle point)
  ‣ Can't initialize all weights to the same value
    - we can show that all hidden units in a layer will always behave the same
    - need to break symmetry
  ‣ Recipe: sample $W_{i,j}^{(k)}$ from $U[-b, b]$ where $b = \frac{\sqrt{6}}{\sqrt{H_k + H_{k-1}}}$
    - the idea is to sample around 0 but break symmetry
    - other values of $b$ could work well (not an exact science) (see Glorot & Bengio, 2010)