Vanishing Gradients

Recall:

\[
\frac{dE}{dW_{ij}} = \sum_k \Delta_k W_{jk} \sigma'(\text{in}_j) (1 - \sigma'(\text{in}_j)) a_i
\]

\[\Delta_j\]

\[
\frac{dE}{dW_{ij}} \to 0 \quad \text{as you go further and further back in the network.}
\]
Weight Initialization

- Heuristic
- Want:
  - mean of in_j to be zero.
  - variance of in_j should be the same across all layers.
- Xavier initialization: For layer l
  \[ W_{ij} \sim N(0, \frac{1}{n_{l-1}}) \]
  where \( n_{l-1} \) is number of nodes in layer \( l-1 \).
- He initialization (effective for ReLU):
  \[ W_{ij} \sim N(0, \frac{2}{n_{l-1}}) \]
Adaptive Step Sizes: \( \alpha_k \) (step size at \( k \)th iteration of GD)

Momentum:

\[ \Delta \theta_k = \eta \Delta \theta_{k-1} + \alpha D L(w_k) \]

\[ \theta_{k+1} = \theta_k - \Delta \theta_k \]

Many more! (Keep a different step size for each model parameter.)

Adaptive gradient (AdaGrad)
Root mean square propagation (RMSProp)
Adaptive moment estimation (Adam)
Classification

Like regression, but $y \in \mathcal{Y}$, where $\mathcal{Y}$ is a set (typically finite) of labels.

Examples: $\mathcal{Y} = \{ \text{is a cat in image, there is no cat} \}$

$\mathcal{Y} = \{ a, b, c, \ldots, z \}$

Idea: One model output per label.

Larger values mean that label is more reasonable

$X \rightarrow \mathcal{Y} \rightarrow \mathcal{Y} \rightarrow \cdots \rightarrow \mathcal{Y} \rightarrow a \rightarrow 0.81$ correct label was $a$. $\rightarrow 0.9 \rightarrow a$
- Predicted label is the largest output.
- Not differentiable. (due to max operator)

**Idea:** Use "softmax" \( e^a \) for last layer.

\[
\Pr(\text{label} = l) = \frac{e^{a_l}}{\sum_l e^{a_l'}}
\]

\[
\Pr(\text{label prediction} = \text{apple}) = \frac{e^{0.9}}{e^{0.9} + e^{0.7} + e^{-1}}
\]

\[
\approx 0.441
\]

\[
\Pr(\text{orange}) \approx 0.36
\]

\[
\Pr(\text{peach}) \approx 0.19
\]
Cross Entropy Loss (Binary)

- Binary Classification, \( |\mathbf{Y}| = 2 \), \( \mathbf{Y} = \{0, 1\} \)

\[
el(\omega) = -\frac{1}{n} \sum_{i=1}^{n} \ln \left( \Pr \left( f_{\omega}(\mathbf{x}_i) = y_i \right) \right)
\]

Monotonic, due to softmax
Generalization bounds

\[ D = n \text{ samples for training.} \]

Data is provided with no explanation. Could be chosen any (not random) way.

Assume data points are sampled i.i.d. from population, independent identically distributed.
\[ l(w) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \] (\(x_i, y_i\) are i.i.d.)

\[ \frac{x_1 + x_2}{2} \]

Hoeffding's inequality:

If \( X_1, \ldots, X_n \) are i.i.d. then

\[ \Pr\left( | \bar{X}_n - \mu | \geq t \right) \leq 2e^{-2nt^2} \]

\[ \frac{x_1 + x_2 + x_3}{3} \]

\[ \bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} x_i \]

\[ \frac{\sum \text{not input } x_i}{\sum} \]

\[ \frac{x_1 \ldots x_{1000}}{1000} \]

\[ \bar{X}_n = 1.7\mu \]

\[ \Pr\left( | \sqrt{1.7n} - \mu | \geq 0.1 \right) \leq 2e^{-2(1000)(0.1)^2} \]

...