Deep Learning

Ulf Brefeld & Soham Majumder

build: May 21, 2024

Machine Learning Group
Leuphana University of Lüneburg

Loss functions

So far, we mainly talked about MSE as the underlying loss function

$$E(f) = \frac{1}{N}\sum_{n=1}^{N}(\overbrace{y_n - f(x_n)}^{\varepsilon \sim \mathcal{N}(0,\sigma^2)})^2$$

MSE is a natural measure for regression problems. For multi-class classification with C classes, target variables $y_n \in \{c_1, \ldots, c_C\}$ can be one-hot encoded by a tensor $z \in \mathbb{R}^{N \times C}$ with

$$\forall n, z_{n,c} = \begin{cases} 1 & \text{if } y_n = c \\ 0 & \text{otherwise} \end{cases}$$

For example, with $N = 4$ and $C = 3$, we obtain

$$\begin{pmatrix} 2 \\ 1 \\ 1 \\ 3 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

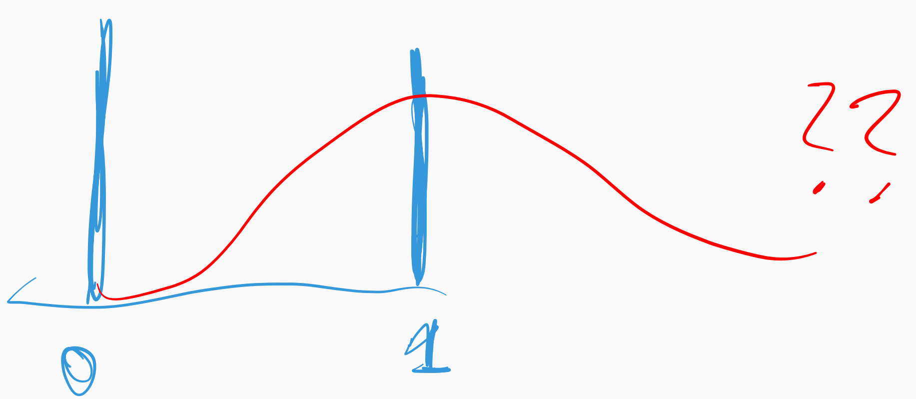
In training, outputs of the model are compared with these binary values using MSE.

However, recall the motivation for MSE:

$$y = f(x) + \epsilon \quad \text{with} \quad \epsilon \sim N(0, \sigma^2) \quad \Rightarrow \quad y - f(x) \sim N(0, \sigma^2)$$

In regression, assuming Gaussian noise around a target value makes sense geometrically. In classification it puts too strong an assumption on the output.

A better choice for a classification loss is the *cross-entropy*.



We begin with binary logistic regression and extend it to the multi-class case. A straightforward way to extend this is to consider the log-linear model for logistic regression.

Cmp. FtS Crammer & Singer

$$\log P(Y = 0|X = x) = w_0 x - \log Z$$
$$\log P(Y = 1|X = x) = w_1 x - \log Z$$

where $-\log Z$ is a “normalizing” term that ensures

$$P(Y = 0|X = x) + P(Y = 1|X = x) = 1$$

and, hence,

$$\frac{1}{Z}\exp(w_0 x) + \frac{1}{Z}\exp(w_1 x) = 1$$
$$\Rightarrow Z = \exp(w_0 x) + \exp(w_1 x).$$

Putting everything together, we get

$$P(Y=0|X=x) = \frac{\exp(w_0 x)}{\exp(w_0 x) + \exp(w_1 x)}, \quad \text{and}$$
$$P(Y=1|X=x) = \frac{\exp(w_1 x)}{\exp(w_0 x) + \exp(w_1 x)}.$$

Note that $P(Y=0|X=x)$ already completely determines the value of $P(Y=1|X=x)$, since both must add up to 1. This implies that once one of them is determined, the other should adapt accordingly. In particular, if we assume $w_1 = 0$,

$$P(Y=1|X=x) = \frac{1}{1+\exp(w_0 x)},$$

which recovers the familiar form of the logistic regression. With C classes, the Z term grows into a sum over all of them.

$$P(Y=y|X=x,W=w)=\frac{1}{Z}\exp f_y(x;w)=\frac{\exp f_y(x;w)}{\sum_c \exp f_c(x;w)}$$

softmax

We have

posterior

likelihood

prior

$$\begin{aligned}
&\log \mu_W(w|\mathcal{D}=\mathbf{d})\\
&=\log\frac{\mu_{\mathcal{D}}(\mathbf{d}|W=w)\mu_W(w)}{\mu_{\mathcal{D}}}\\
&=\log \mu_{\mathcal{D}}(\mathbf{d}|W=w)+\log \mu_W(w)-\log Z'\\
&=\sum_{n=1}^{N}\log \mu_{\mathcal{D}}(x_n,y_n|W=w)+\log \mu_W(w)-\log Z'\\
&=\sum_n \log P(Y=y_n|X=x_n,W=w)+\log \mu_W(w)-\log Z'\\
&=\underbrace{\sum_n \log\left(\frac{\exp f_y(x;w)}{\sum_c \exp f_c(x;w)}\right)}_{\text{depends on outputs}}+\underbrace{\log \mu_W(w)}_{\text{depends on } w}-\log Z'
\end{aligned}$$

$\mu_{\mathcal{D}} \longrightarrow$ partition function

iid

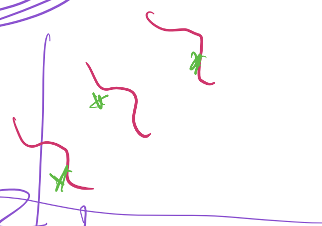
$$p(w) \sim N(0, \sigma^2)$$

$$\sim \frac{1}{(2\pi\Sigma)^{D/2}} \exp\left\{ \frac{-1\|w-0\|^2}{2\,\Sigma^{-1}} \right\}$$

$$\log p(w) \Rightarrow \underbrace{\log \frac{1}{\cdots}}_{\text{constant}} - \|w\|^2 \cdot \underbrace{\frac{1}{2\sigma^2}}_{\text{constant}}$$

$$\log p(w \mid D) = \log\left(\frac{p(D \mid w)\, p(w)}{p(D)} \right)$$

$$\underbrace{y_n - f(x_n)} = \varepsilon_n \sim N(0, \sigma^2)$$



$$\log p(D \mid w) = \sum_{n=1}^{N} \underbrace{\log\left(\frac{1}{(2\pi\sigma)^{D/2}} \right)}_{\text{constant}} \exp\left\{ \frac{\|y - f(x)\|^2}{-2\Sigma^{-1}} \right\}$$

$$= \sum_{n=1}^{N} -\|y - f(x)\|^2$$

Under MAP (maximum-a-posteriori)

$$\arg\max_w P(w|D) = \sum_{n=1}^{N} - \|y - f(x)\|^2 - \|w\|^2$$

Started: $P(w|D) = \dfrac{P(D|w)\,P(w)}{P(D)}$

for regression and linear model
$f(x) = w^T x$

$$\arg\max_w \log P(w|D) \propto \underbrace{-\sum_{n=1}^{N} \|y_n - f(x_n)\|^2}_{\text{likelihood}} \underbrace{- \|w\|^2}_{\text{prior}}$$

$$\Leftrightarrow \arg\min_w \underbrace{\sum \|y - f(x)\|^2}_{\text{loss function}} + \underbrace{\|w\|^2}_{\text{Regularizer}}$$

---

assumption in regression:

$y = f(x) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

$$\varepsilon \sim \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{\|\cancel{\varepsilon}\ y - f(x)\|^2}{2\sigma^2} \right\}$$

$$p(w \mid D) = p(w \mid \{(x_i, y_i)\}_{i=1}^{N})$$

$$= \frac{p(y \mid w, x)\, p(w)}{p(D)}$$

If we ignore the prior/regularizer on w, we could minimize

$$E(w) = -\frac{1}{N}\sum_n \log \underbrace{\left(\frac{\exp f_y(x;w)}{\sum_c \exp f_c(x;w)}\right)}_{\hat{P}(Y=y|X=x)}$$

Given two distributions p and q, their **cross-entropy** is defined as

$$\mathbb{H}(p,q) = -\sum_k p(k)\log q(k)$$

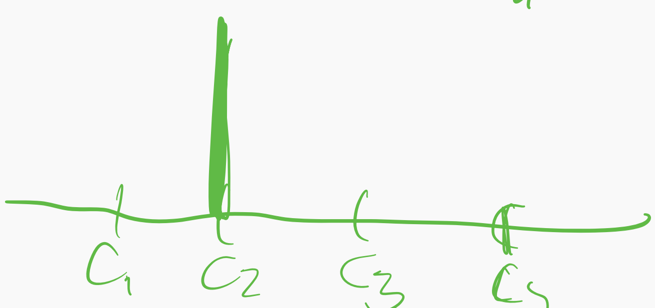
$$\delta_y(c) := \begin{cases} 1 : y=c \\ 0 : \text{otherwise} \end{cases}$$

using $0\log 0 = 0$. We obtain

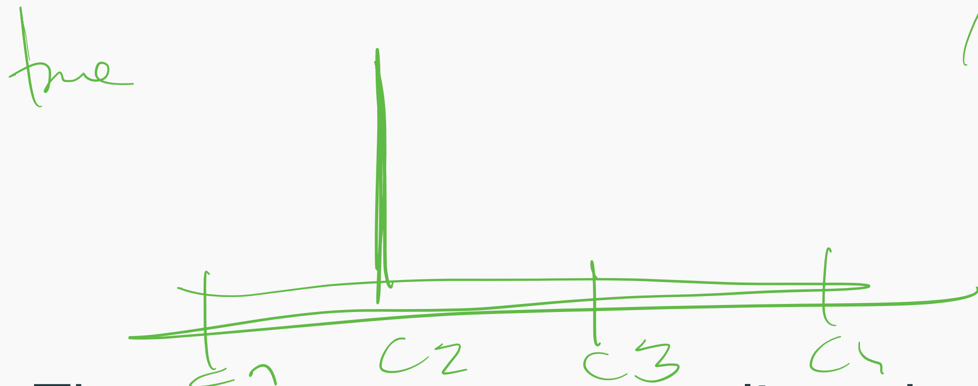
$$-\log\left(\frac{\exp f_{y_n}(x_n;w)}{\sum_c \exp f_c(x_n;w)}\right) = -\log \hat{P}_w(Y = y_n|X = X_n)$$

$$= -\sum_c \delta_{y_n}(c)\log \hat{P}_w(Y = c|X = x_n)$$

$$= \mathbb{H}(\delta_{y_n}, \hat{P}(Y = \cdot\,|X = x_n))$$



E is the mean cross-entropy between the true posterior δ_{y_n} and the estimate $\hat{P}(Y = \cdot\,|X = x_n)$



The cross-entropy normalizes the decision values (or net outputs) into logs of probabilities, e.g.

$$(\alpha_1, \ldots, \alpha_C) \mapsto \left(\log \frac{\exp \alpha_1}{\sum_c \exp \alpha_c}, \ldots, \log \frac{\exp \alpha_C}{\sum_c \exp \alpha_c} \right)$$

The softmax has the notion of a winner-takes-all, making the largest output even larger and others smaller

Vanishing Gradients

Recall the gradients for an MLP.

Forward pass

$$s^{(\ell)} = w^{(\ell)} x^{(\ell-1)} + b^{(\ell)}$$
$$x^{(\ell)} = \sigma(s^{(\ell)})$$

with $x^{(0)} = x$, $\forall \ell = 1, \dots, L$

Backward pass

$$\left[\!\!\left[\frac{\partial E}{\partial w^{(\ell)}} \right]\!\!\right] = \left[\frac{\partial E}{\partial s^{(\ell)}} \right] \left(x^{(\ell-1)} \right)^\top \quad \text{and} \quad \left[\frac{\partial E}{\partial b^{(\ell)}} \right] = \left[\frac{\partial E}{\partial s^{(\ell)}} \right]$$

Recall the gradients for an MLP.

Forward pass

$$s^{(\ell)} = w^{(\ell)} x^{(\ell-1)} + b^{(\ell)}$$

$$x^{(\ell)} = \sigma\left(s^{(\ell)}\right)$$

with $x^{(0)} = x$, $\forall \ell = 1, \dots, L$

Backward pass

$$\left[\!\left[\frac{\partial E}{\partial w^{(\ell)}}\right]\!\right] = \left[\frac{\partial E}{\partial s^{(\ell)}}\right] \left(x^{(\ell-1)}\right)^\top \quad \text{and} \quad \left[\frac{\partial E}{\partial b^{(\ell)}}\right] = \left[\frac{\partial E}{\partial s^{(\ell)}}\right]$$

$$\left[\frac{\partial E}{\partial s^{(\ell)}}\right] = \left[\frac{\partial E}{\partial x^{(\ell)}}\right] \odot \sigma'\left(s^{(\ell)}\right)$$

Recall the gradients for an MLP.

Forward pass

$$s^{(\ell)} = w^{(\ell)} x^{(\ell-1)} + b^{(\ell)}$$

$$x^{(\ell)} = \sigma(s^{(\ell)})$$

with $x^{(0)} = x$, $\forall \ell = 1, \dots, L$

Backward pass

$$\left[\!\left[\frac{\partial E}{\partial w^{(\ell)}}\right]\!\right] = \left[\frac{\partial E}{\partial s^{(\ell)}}\right]\left(x^{(\ell-1)}\right)^\top \quad \text{and} \quad \left[\frac{\partial E}{\partial b^{(\ell)}}\right] = \left[\frac{\partial E}{\partial s^{(\ell)}}\right]$$

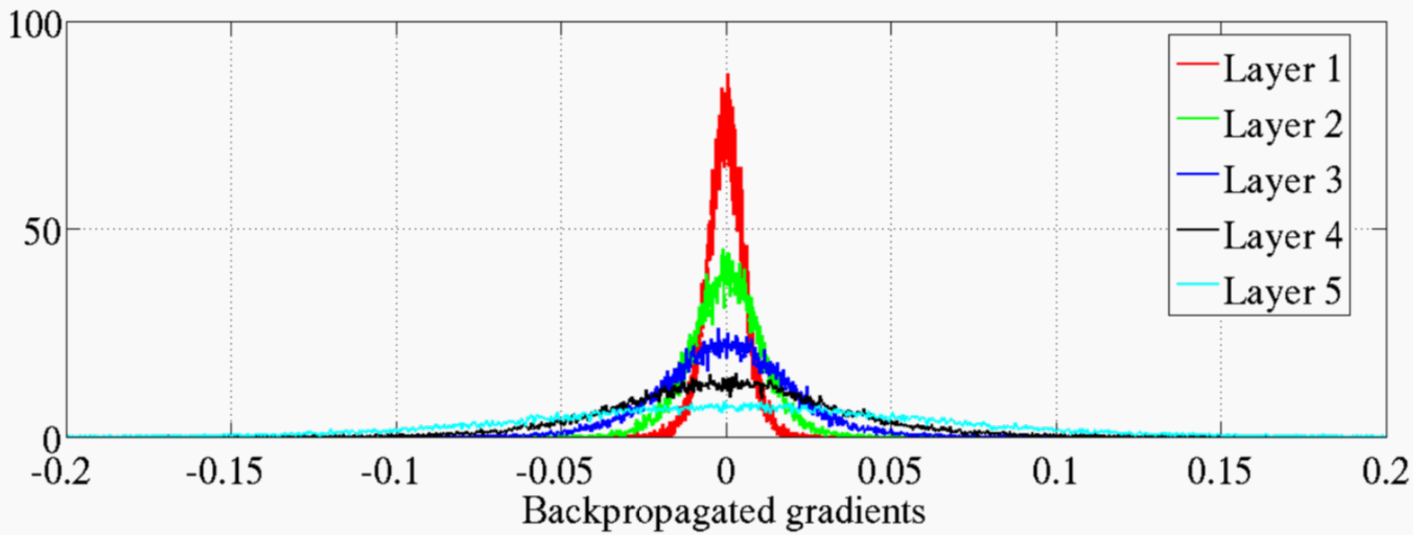
$$\left[\frac{\partial E}{\partial s^{(\ell)}}\right] = \left[\frac{\partial E}{\partial x^{(\ell)}}\right] \odot \sigma'(s^{(\ell)})$$

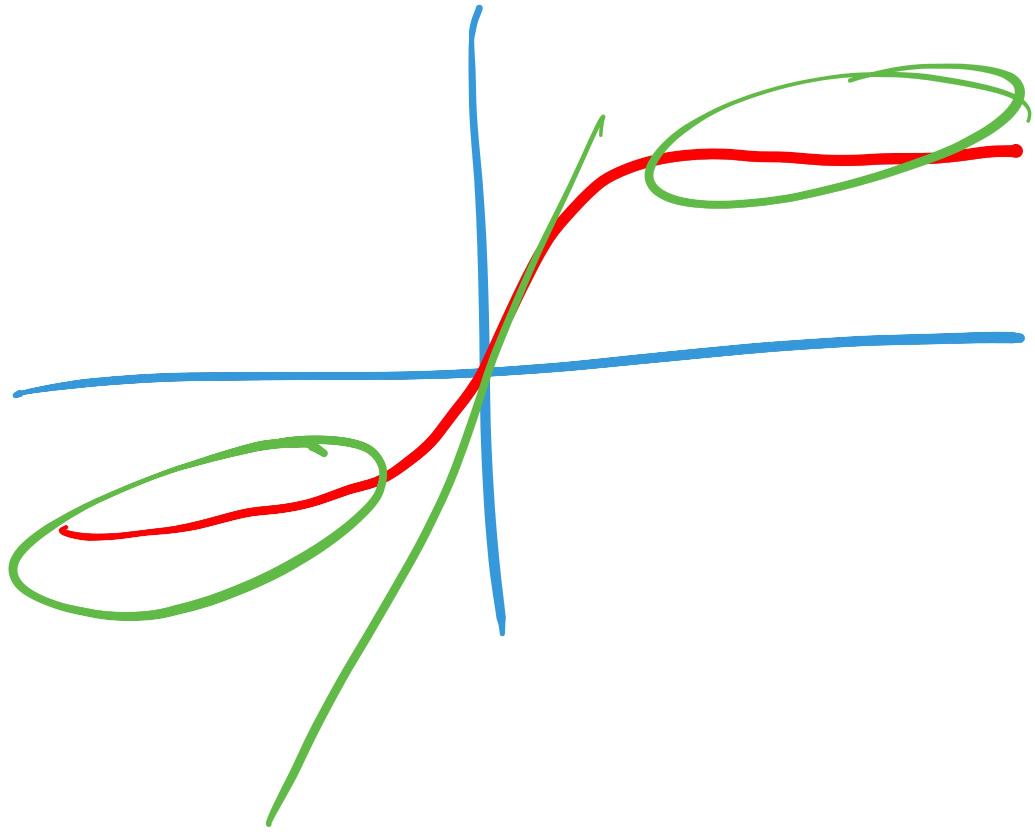
$$\left[\frac{\partial E}{\partial x^{(L)}}\right] = \nabla_1 E(x^{(L)}) \quad \text{and} \quad \left[\frac{\partial E}{\partial x^{(\ell)}}\right] = \left(w^{(\ell+1)}\right)^\top \left[\frac{\partial E}{\partial s^{(\ell+1)}}\right], \quad \forall \ell < L$$

We thus have

$$\left[\frac{\partial E}{\partial x^{(\ell)}}\right] = \left(w^{(\ell+1)}\right)^{\top} \overbrace{\left(\left[\frac{\partial E}{\partial x^{(\ell+1)}}\right] \odot \sigma'\left(s^{(\ell+1)}\right)\right)}^{\left[\frac{\partial E}{\partial s^{(\ell+1)}}\right]}$$

A problem is that the gradient 'vanishes' exponentially (in depth ℓ), if the w's are ill-conditioned or the activations are in the saturating domain of σ (Glorot & Bengio, 2010)





Idea: need to control the variances

$$Var\left(\frac{\partial E}{\partial w_{i,j}^{(\ell)}}\right) \quad \text{and} \quad Var\left(\frac{\partial E}{\partial b_i^{(\ell)}}\right)$$

such that

- the gradient does not vanish
- weights evolve at the same rate across layers during training, no layer saturates before others

Recall, if two random variables A and B are independent then

$$\begin{aligned} Var(AB) &= \mathbb{E}(A^2B^2) - (\mathbb{E}(AB))^2 \\ &= Var(A)Var(B) + Var(A)\mathbb{E}(B)^2 + Var(B)\mathbb{E}(A)^2 \end{aligned}$$

To not clutter notation unnecessarily, we will drop indexes when variances are identical for all activations/parameters in a layer

Consider the ℓ-th layer with N_ℓ units and activation function σ,

$$x_i^{(\ell)} = \sigma\left(\sum_{j=1}^{N_{\ell-1}} w_{i,j}^{(\ell)} x_j^{(\ell-1)} + b_i^{(\ell)}\right)$$

Assume $\sigma'(0) = 1$ and that $x_i^{(\ell)}$ is a linear function

$$x_i^{(\ell)} \approx \sum_{j=1}^{N_{\ell-1}} w_{i,j}^{(\ell)} x_j^{(\ell-1)} + b_i^{(\ell)}$$

with $w^{(l)}$ and $x^{(\ell-1)}$ centered, then

$$Var(x_i^{(\ell)}) \approx Var\left(\sum_{j=1}^{N_{\ell-1}} w_{i,j}^{(\ell)} x_j^{(\ell-1)}\right)$$

$$= \sum_{j=1}^{N_{\ell-1}} Var\left(w_{i,j}^{(\ell)}\right) Var\left(x_j^{(\ell-1)}\right)$$

If $w_{i,j}^{(\ell)}$ are i.i.d. and the $x_i^{(\ell)}$ have the same variance in layer ℓ

$$Var\left(x^{(\ell)}\right) \approx \sum_{j=1}^{N_{\ell-1}} Var\left(w_{i,j}^{(\ell)}\right) Var\left(x_j^{(\ell-1)}\right)$$

$$= N_{\ell-1} Var\left(w^{(\ell)}\right) Var\left(x^{(\ell-1)}\right)$$

The **variance of the activations** is then given by

$$Var\left(x^{(\ell)}\right) \approx Var\left(x^{(0)}\right) \prod_{k=1}^{\ell} N_{k-1} Var\left(w^{(k)}\right)$$

Hence, a possible initialization fulfills

$$Var\left(w^{(\ell)}\right) = \frac{1}{N_{\ell-1}}$$

!

$w^T x + b = \sum_{d=1}^{D} w_d x_d + b$

We now consider the variance of the gradient wrt the activations. From

$$
\begin{aligned}
\frac{\partial E}{\partial x_i^{(\ell)}} &= \sum_{k=1}^{N_{\ell+1}} \frac{\partial E}{\partial x_k^{(\ell+1)}} \frac{\partial x_k^{(\ell+1)}}{\partial x_i^{(\ell)}} \\
&= \sum_{k=1}^{N_{\ell+1}} \frac{\partial E}{\partial x_k^{(\ell+1)}} w_{k,i}^{(\ell+1)}
\end{aligned}
$$

we obtain

$$
Var\left(\frac{\partial E}{\partial x^{(\ell)}}\right) \approx N_{\ell+1} Var\left(\frac{\partial E}{\partial x^{(\ell+1)}}\right) Var\left(w^{(\ell+1)}\right)
$$

For the **variance of the gradients wrt activations**, it holds

$$
Var\left(\frac{\partial E}{\partial x^{(\ell)}}\right) \approx Var\left(\frac{\partial E}{\partial x^{(L)}}\right) \prod_{k=\ell+1}^{L} N_k Var(w^{(k)})
$$

Since

$$x_i^{(\ell)} \approx \sum_{j=1}^{N_{\ell-1}} w_{i,j}^{(\ell)} x_j^{(\ell-1)} + b_i^{(\ell)}$$

we have

$$\frac{\partial E}{\partial w_{i,j}^{(\ell)}} = \frac{\partial E}{\partial x_i^{(\ell)}} \frac{\partial x_i^{(\ell)}}{\partial w_{i,j}^{(\ell)}} \approx \frac{\partial E}{\partial x_i^{(\ell)}} x_j^{(\ell-1)}$$

The **variance of the gradient wrt the weights** is given by

$N_{\ell+1} \dots N_{\ell+2} \dots N_L$

$$\begin{aligned}
Var\left(\frac{\partial E}{\partial w^{(\ell)}}\right) &\approx Var\left(\frac{\partial E}{\partial x^{(\ell)}}\right) Var\left(x^{(\ell)}\right) \\
&= Var\left(\frac{\partial E}{\partial x^{(L)}}\right) \left(\prod_{k=\ell+1}^{L} N_k Var\left(w^{(k)}\right)\right) Var\left(x^{(0)}\right) \cdot \\
&\quad \cdot \left(\prod_{k=1}^{\ell} N_{k-1} Var\left(w^{(k)}\right)\right) \\
&= \frac{N_0}{N_\ell} \left(\prod_{k=1}^{L} N_k Var\left(w^{(k)}\right)\right) Var\left(x^{(0)}\right) Var\left(\frac{\partial E}{\partial x^{(L)}}\right)
\end{aligned}$$

$N_0 \, N_1 \dots N_{\ell-1}$

Similarly, for the biases we obtain

$$x_i^{(\ell)} \approx \sum_{j=1}^{N_{\ell-1}} w_{i,j}^{(\ell)} x_j^{(\ell-1)} + b_i^{(\ell)}$$

and hence,

$$\frac{\partial E}{\partial b_i^{(\ell)}} = \frac{\partial E}{\partial x_i^{(\ell)}} \frac{\partial x_i^{(\ell)}}{\partial b_i^{(\ell)}} \approx \frac{\partial E}{\partial x_i^{(\ell)}}$$

The **variance of the gradient wrt the biases** is given by

$$Var\left(\frac{\partial E}{\partial b^{(\ell)}}\right) \approx Var\left(\frac{\partial E}{\partial x^{(\ell)}}\right)$$

Thus, there is nothing we could do to effectively control the variance of the gradient wrt the weights

The variance of activations can be controlled by

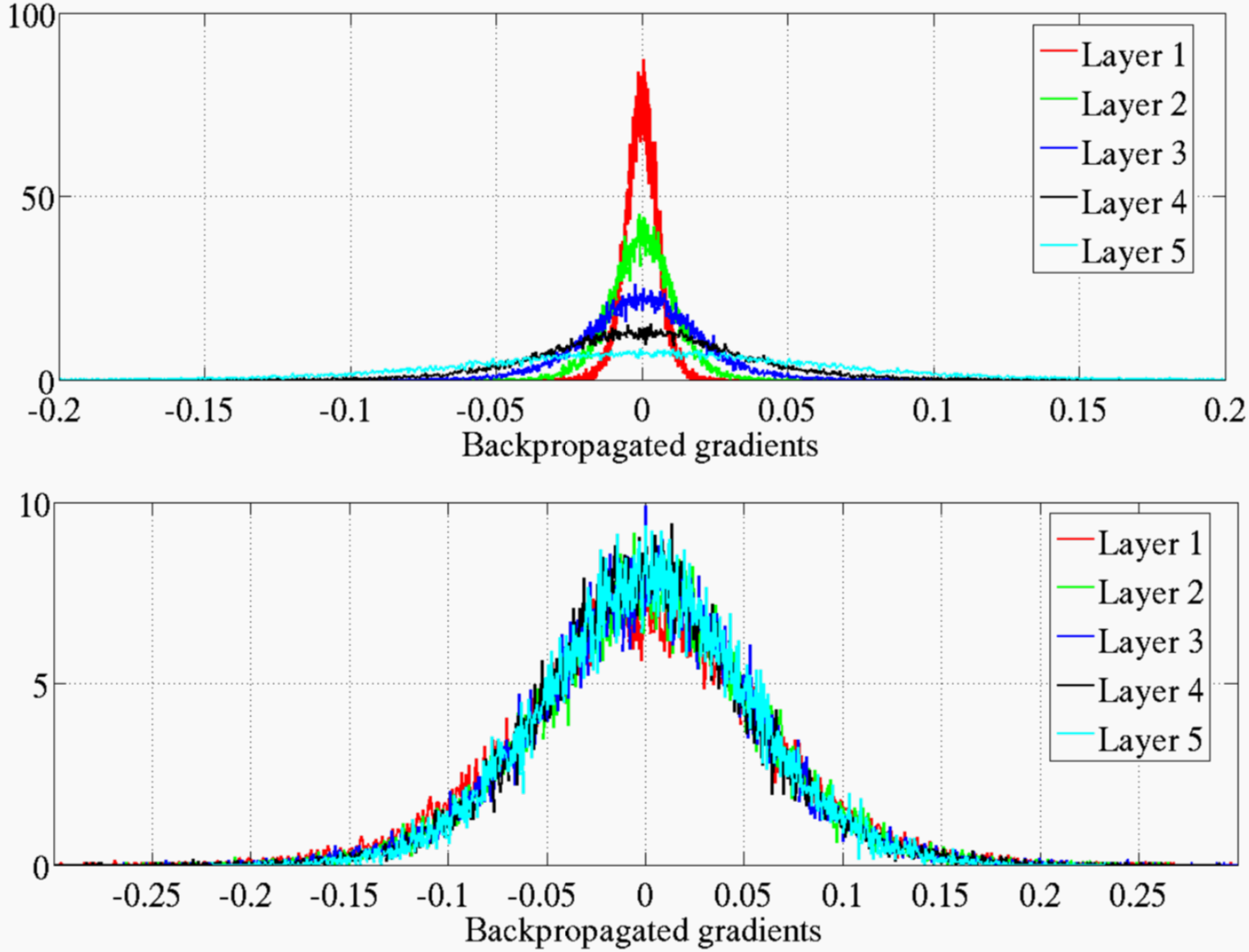
$$Var(w^{(\ell)}) = \frac{1}{N_{\ell-1}}$$

and to control the variance of the gradient wrt activations (and intrinsically also the variance of the gradient wrt the biases) we set

$$Var(w^{(\ell)}) = \frac{1}{N_{\ell}}$$

The so-called 'Xavier initialization' serves as a compromise between the two and is given by (Glorot & Bengio, 2010)

$$Var(w^{(\ell)}) = \frac{1}{\frac{N_{\ell-1}+N_{\ell}}{2}} = \frac{2}{N_{\ell-1} + N_{\ell}}$$



(Glorot & Bengio, 2010)

Weights can also be scaled to account for the activation:

Recall that we have

$$\begin{aligned}\mathbb{E}\left[s^{(\ell)}\right] &= \mathbb{E}\left[w^{(\ell)} \cdot x^{(\ell)} + b^{(\ell)}\right] \\ &= \underbrace{\mathbb{E}\left[w^{(\ell)}\right]}_{\text{=0 (assumed)}} \mathbb{E}\left[x^{(\ell)}\right] + \underbrace{\mathbb{E}\left[b^{(\ell)}\right]}_{\text{=0 (assumed)}} = 0,\end{aligned}$$

and $s^{(\ell-1)}$ is symmetric. For a ReLu activation σ

$$\begin{aligned}\mathbb{E}\left[\sigma(s)^2\right] &= \int_{-\infty}^{+\infty} \max(0,s)^2 p(s)ds \\ &= \int_{0}^{+\infty} s^2 p(s)ds \\ &= \frac{1}{2}\int_{-\infty}^{+\infty} s^2 p(s)ds \\ &= \frac{1}{2}\int_{-\infty}^{+\infty} (s - \mathbb{E}[s])^2 p(s)ds \\ &= \frac{1}{2}\mathbb{E}\left[(s-\mathbb{E}[s])^2\right] = \frac{1}{2}Var(s)\end{aligned}$$

Assume the foward pass is given by

$$s_i^{(\ell)} = \sum_{j=1}^{N_{\ell-1}} w_{i,j}^{(\ell)} \sigma\left(s_j^{(\ell-1)}\right) + b_i^{(\ell)} \quad \text{and} \quad x_i^{(\ell)} = \sigma(s_i^{(\ell)})$$

then,

$$\begin{aligned} Var\left(s_i^{(\ell)}\right) &= N_{\ell-1} Var\left(w^{(\ell)} \sigma\left(s^{(\ell-1)}\right)\right) \\ &= N_{\ell-1} Var\left(w^{(\ell)}\right) \mathbb{E}\left(\sigma\left(s^{(\ell-1)}\right)^2\right) \\ &= \frac{1}{2} N_{\ell-1} Var\left(w^{(\ell)}\right) Var\left(s^{(\ell-1)}\right) \end{aligned}$$

For the backward pass, we have

$$
\begin{aligned}
Var\left(\frac{\partial E}{\partial x_i^{(\ell)}}\right) &= \sum_{h=1}^{N_{\ell+1}} Var\left(\underbrace{\sigma'\left(s_h^{(\ell+1)}\right)}_{0/1} \underbrace{\frac{\partial E}{\partial x_h^{(\ell+1)}} w_{h,i}^{(\ell+1)}}_{\mathbb{E}(\cdot)=0,\,\text{symmetric}}\right) \\
&= \sum_{h=1}^{N_{\ell+1}} \mathbb{E}\left(\sigma'\left(s_h^{(\ell+1)}\right)\left(\frac{\partial E}{\partial x_h^{(\ell+1)}} w_{h,i}^{(\ell+1)}\right)^2\right) \\
&= \sum_{h=1}^{N_{\ell+1}} \frac{1}{2}\mathbb{E}\left(\left(\frac{\partial E}{\partial x_h^{(\ell+1)}} w_{h,i}^{(\ell+1)}\right)^2\right) \\
&= \frac{1}{2}\sum_{h=1}^{N_{\ell+1}} Var\left(\frac{\partial E}{\partial x_h^{(\ell+1)}}\right) Var\left(w_{h,i}^{(\ell+1)}\right)
\end{aligned}
$$

Thus, the effect of ReLU on the forward and backward pass is as if the weights had only half the variance, which motivates multiplying them by a corretive gain of $\sqrt{2}$ (He et al., 2015)

Normalizing the data

The analysis of the weight initialization relies on a constant variance of the activations.

For this to be true, not only the variance has to remain unchanged through layers, but it has to be correct for the input too

$$Var\left(x^{(0)}\right) = 1$$

$$x_n = (100000, 0.0001)^T$$

$$\sigma(w^T x + b) = \ldots$$

References

X. Glorot & Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings opf the International Conference on Artificial Intelligence and Statistics, 2010.

K. He, X. Zhang, S. Ren & J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR, abs/1502.01852, 2015.