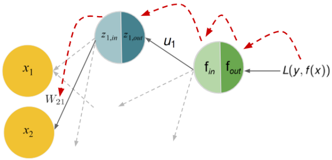
Deep Learning

Basic Backpropagation 1



Learning goals

- Forward and backward passes
- Chain rule
- Details of backprop

BACKPROPAGATION: BASIC IDEA

We would like to optimize ERM using gradient descent (GD) on:

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right).$$

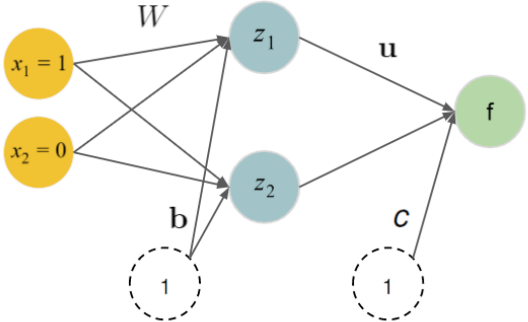
Backprop training of NNs runs in 2 alternating steps, for one **x**:

1. **Forward pass (FP):** Inputs flow through model to outputs. We then compute the observation loss (see previous chapters).
2. **Backward pass (BP):** Loss flows backwards to update weights so error is reduced, as in GD.

We will see: This is simply (S)GD in disguise, cleverly using the chain rule, so we can reuse a lot of intermediate results.

XOR EXAMPLE

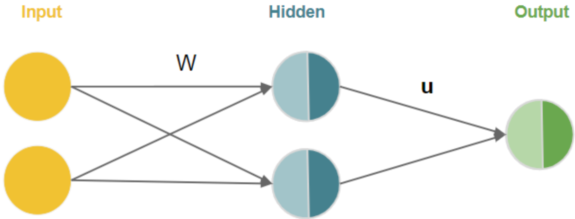
- As activations (hidden and outputs) we use the sigmoid function.
- We run one FP and BP on $\mathbf{x} = (1, 0)^T$ with $y = 1$.
- We use L_2 loss between 0-1 labels and the predicted probabilities. This is a bit uncommon, but computations become simpler.



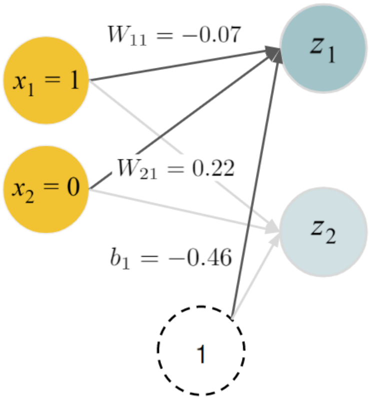
Note: We will only show rounded decimals.

FORWARD PASS

- We will divide the FP into four steps:
  - the inputs of z_i: $\boldsymbol{z_{i,in}}$
  - the activations of z_i: $\boldsymbol{z_{i,out}}$
  - the input of f: $\boldsymbol{f_{in}}$
  - and finally the activation of f: $\boldsymbol{f_{out}}$

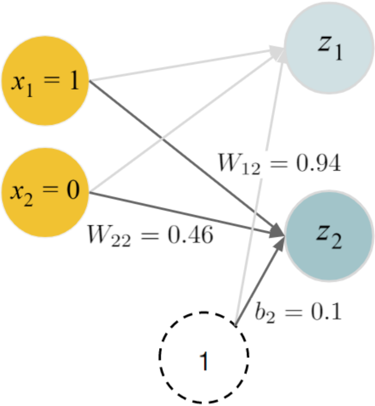


FORWARD PASS



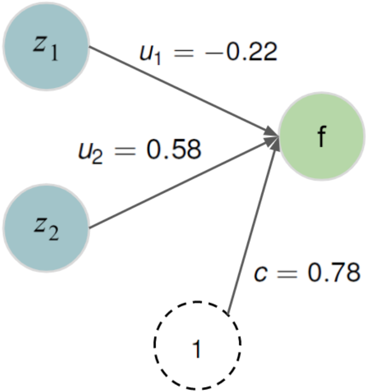
$$
\begin{aligned}
z_{1,in} &= \mathbf{W}_1^T\mathbf{x} + b_1 = 1 \cdot (-0.07) + 0 \cdot 0.22 + 1 \cdot (-0.46) = -0.53 \\
z_{1,out} &= \sigma\left(z_{1,in}\right) = \frac{1}{1 + \exp(-(-0.53))} = 0.3705
\end{aligned}
$$

FORWARD PASS



$$
\begin{aligned}
z_{2,in} &= \mathbf{W}_2^T \mathbf{x} + b_2 = 1 \cdot 0.94 + 0 \cdot 0.46 + 1 \cdot 0.1 = 1.04 \\
z_{2,out} &= \sigma\left(z_{2,in}\right) = \frac{1}{1 + \exp(-1.04)} = 0.7389
\end{aligned}
$$

FORWARD PASS



$$
\begin{aligned}
f_{in} &= \boldsymbol{u}^T \boldsymbol{z} + c = 0.3705 \cdot (-0.22) + 0.7389 \cdot 0.58 + 1 \cdot 0.78 = 1.1122 \\
f_{out} &= \tau\left(f_{in}\right) = \frac{1}{1 + \exp(-1.1122)} = 0.7525
\end{aligned}
$$

FORWARD PASS

- The FP predicted $f_{out} = 0.7525$
- Now, we compare the prediction $f_{out} = 0.7525$ and the true label $y = 1$ using the L_2-loss:

$$\begin{aligned} L\left(y, f(\mathbf{x})\right) &= \frac{1}{2}\left(y - f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)^2 = \frac{1}{2}\left(y - f_{out}\right)^2 \\ &= \frac{1}{2}\left(1 - 0.7525\right)^2 = 0.0306 \end{aligned}$$

- The calculation of the gradient is performed backwards (starting from the output layer), so that results can be reused.

BACKWARD PASS

The main ingredients of the backward pass are:

- to reuse the results of the forward pass
  (here: $z_{i,in}$, $z_{i,out}$, f_{in}, f_{out})
- reuse the intermediate results from the chain rule
- the derivative of the activations and some affine functions

BACKWARD PASS

- Let's start to update u_1. We recursively apply the chain rule:

$$\frac{\partial L\left(y, f(\mathbf{x})\right)}{\partial u_1} = \frac{\partial L\left(y, f(\mathbf{x})\right)}{\partial f_{out}} \cdot \frac{\partial f_{out}}{\partial f_{in}} \cdot \frac{\partial f_{in}}{\partial u_1}$$

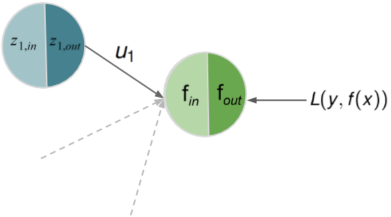


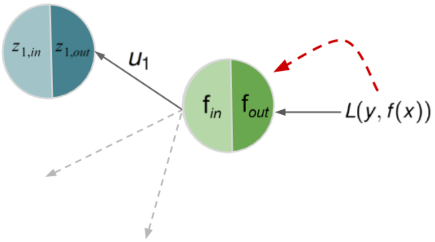
Figure: Snippet from our NN, with backward path for u_1.

BACKWARD PASS

- 1st step: The derivative of L_2 loss is easy; we know f_{out} from FP.

$$\frac{\partial L(y, f(\mathbf{x}))}{\partial f_{out}} = \frac{d}{\partial f_{out}} \frac{1}{2}(y - f_{out})^2 = -\underbrace{(y - f_{out})}_{\hat{=}\text{residual}}$$

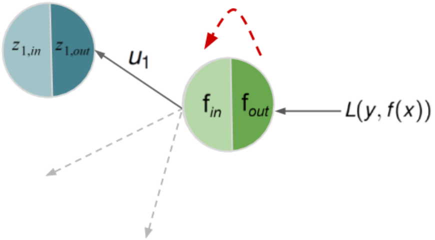
$$= -(1 - 0.7525) = -0.2475$$



BACKWARD PASS

- 2nd step. $f_{out} = \sigma(f_{in})$, use rule for σ', use f_{in} from FP.

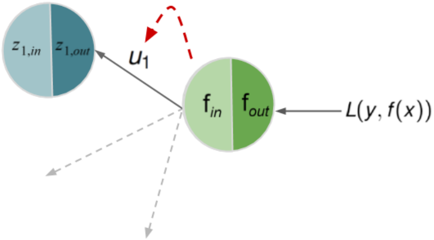
$$
\begin{aligned}
\frac{\partial f_{out}}{\partial f_{in}} &= \sigma(f_{in}) \cdot (1 - \sigma(f_{in})) \\
&= 0.7525 \cdot (1 - 0.7525) = 0.1862
\end{aligned}
$$



BACKWARD PASS

- 3rd step. Derivative of the linear input is easy; use $z_{1,out}$ from FP.

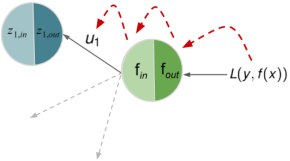
$$\frac{\partial f_{in}}{\partial u_1} = \frac{\partial(u_1 \cdot z_{1,out} + u_2 \cdot z_{2,out} + c \cdot 1)}{\partial u_1} = z_{1,out} = 0.3705$$



BACKWARD PASS

- Plug it together:

$$\begin{aligned}\frac{\partial L(y, f(\mathbf{x}))}{\partial u_1} &= \frac{\partial L(y, f(\mathbf{x}))}{\partial f_{out}} \cdot \frac{\partial f_{out}}{\partial f_{in}} \cdot \frac{\partial f_{in}}{\partial u_1} \\ &= -0.2475 \cdot 0.1862 \cdot 0.3705 = -0.0171\end{aligned}$$



- With LR $\alpha = 0.5$:

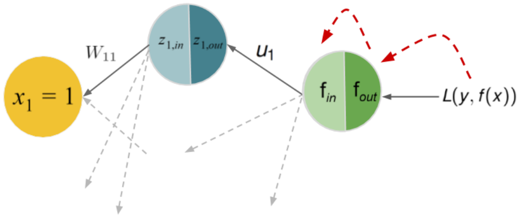
$$\begin{aligned}u_1^{[new]} &= u_1^{[old]} - \alpha \cdot \frac{\partial L(y, f(\mathbf{x}))}{\partial u_1} \\ &= -0.22 - 0.5 \cdot (-0.0171) = -0.2115\end{aligned}$$

BACKWARD PASS

- Now for W_{11}:

$$\frac{\partial L(y, f(\mathbf{x}))}{\partial W_{11}} = \frac{\partial L(y, f(\mathbf{x}))}{\partial f_{out}} \cdot \frac{\partial f_{out}}{\partial f_{in}} \cdot \frac{\partial f_{in}}{\partial z_{1,out}} \cdot \frac{\partial z_{1,out}}{\partial z_{1,in}} \cdot \frac{\partial z_{1,in}}{\partial W_{11}}$$

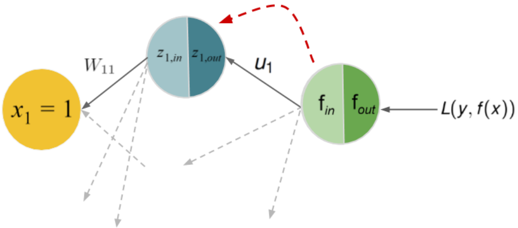
- We know $\frac{\partial L(y,f(\mathbf{x}))}{\partial f_{out}}$ and $\frac{\partial f_{out}}{\partial f_{in}}$ from BP for u_1.



BACKWARD PASS

- $f_{in} = u_1 \cdot z_{1,out} + u_2 \cdot z_{2,out} + c \cdot 1$ is linear, easy and we know u_1 :

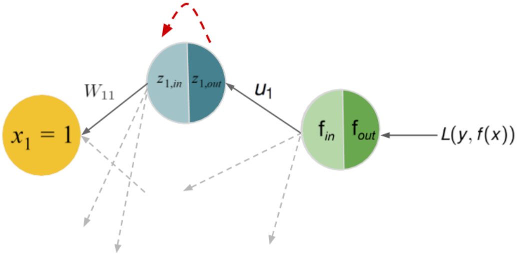
$$\frac{\partial f_{in}}{\partial z_{1,out}} = u_1 = -0.22$$



BACKWARD PASS

- Next. Use rule for σ' and FP results:

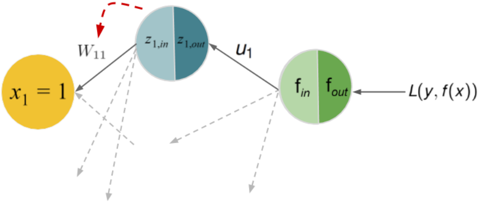
$$\begin{aligned}\frac{\partial z_{1,out}}{\partial z_{1,in}} &= \sigma(z_{1,in}) \cdot (1 - \sigma(z_{1,in})) \\ &= 0.3705 \cdot (1 - 0.3705) = 0.2332\end{aligned}$$



BACKWARD PASS

- $z_{1,in} = x_1 \cdot W_{11} + x_2 \cdot W_{21} + b_1 \cdot 1$ is linear and depends on inputs:

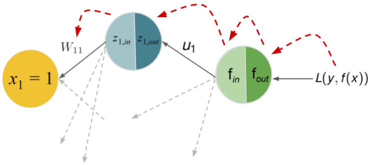
$$\frac{\partial z_{1,in}}{\partial W_{11}} = x_1 = 1$$



BACKWARD PASS

- Plugging together:

$$
\begin{aligned}
\frac{\partial L(y, f(\mathbf{x}))}{\partial W_{11}} &= \frac{\partial L(y, f(\mathbf{x}))}{\partial f_{out}} \cdot \frac{\partial f_{out}}{\partial f_{in}} \cdot \frac{\partial f_{in}}{\partial z_{1,out}} \cdot \frac{\partial z_{1,out}}{\partial z_{1,in}} \cdot \frac{\partial z_{1,in}}{\partial W_{11}} \\
&= (-0.2475) \cdot 0.1862 \cdot (-0.22) \cdot 0.2332 \cdot 1 \\
&= 0.0024
\end{aligned}
$$



- Full SGD update:

$$
\begin{aligned}
W_{11}^{[new]} &= W_{11}^{[old]} - \alpha \cdot \frac{\partial L(y, f(\mathbf{x}))}{\partial W_{11}} \\
&= -0.07 - 0.5 \cdot 0.0024 = -0.0712
\end{aligned}
$$

RESULT

- We can do this for all weights:

$$W = \begin{pmatrix} -0.0712 & 0.9426 \\ 0.22 & 0.46 \end{pmatrix}, b = \begin{pmatrix} -0.4612 \\ 0.1026 \end{pmatrix},$$

$$u = \begin{pmatrix} -0.2115 \\ 0.5970 \end{pmatrix} \text{ and } c = 0.8030.$$

- Yields $f(\mathbf{x} \mid \theta^{[new]}) = 0.7615$ and loss $\frac{1}{2}(1 - 0.7615)^2 = 0.0284$.
- Before, we had $f(\mathbf{x} \mid \theta^{[old]}) = 0.7525$ and higher loss 0.0306.

Now rinse and repeat. This was one training iter, we do thousands.