

# Practical-1: Pen and paper exercises

Georgios Methenitis

November 9, 2016

## Exercise 1

We start by the loss function which is given by  $\mathcal{L} = 0.5(y_{out} - y_{gt})^2$ . Using this we compute the derivatives of the loss function with respect to the weights starting from

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_{out}} &= \frac{\partial}{\partial W_{out}} (0.5(y_{out} - y_{gt})^2) \\ &= 2 \times 0.5(y_{out} - y_{gt}) \frac{\partial}{\partial W_{out}} (y_{out} - y_{gt}) \\ &= (y_{out} - y_{gt}) \frac{\partial}{\partial W_{out}} y_{out} \\ &= (y_{out} - y_{gt}) \frac{\partial}{\partial W_{out}} (f_3(W_{out} f_2(w_2 f_1(w_1 x_{in})))) , \text{ by the chain rule} \\ &= (y_{out} - y_{gt}) \frac{\partial f_3(s_{out})}{\partial W_{out}} z_2.\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_2} &= (y_{out} - y_{gt}) \frac{\partial \mathcal{L}}{\partial W_2} (f_3(W_{out} f_2(w_2 f_1(w_1 x_{in})))) , \text{ by the chain rule} \\ &= (y_{out} - y_{gt}) \frac{\partial f_3(s_{out})}{\partial W_{out}} \frac{\partial \mathcal{L}}{\partial W_2} (f_2(w_2 f_1(w_1 x_{in}))) \\ &= (y_{out} - y_{gt}) \frac{\partial f_3(s_{out})}{\partial W_{out}} \frac{\partial f_2(s_2)}{\partial W_2} z_1.\end{aligned}$$

And similarly,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_1} &= (y_{out} - y_{gt}) \frac{\partial \mathcal{L}}{\partial W_1} (f_3(W_{out} f_2(w_2 f_1(w_1 x_{in})))) , \text{ by the chain rule} \\ &= (y_{out} - y_{gt}) \frac{\partial f_3(s_{out})}{\partial W_{out}} \frac{\partial f_2(s_2)}{\partial W_2} \frac{\partial \mathcal{L}}{\partial W_1} (f_1(w_1 x_{in})) \\ &= (y_{out} - y_{gt}) \frac{\partial f_3(s_{out})}{\partial W_{out}} \frac{\partial f_2(s_2)}{\partial W_2} \frac{\partial f_1(s)}{\partial W_1}.\end{aligned}$$

## Prelude

We start by  $\Delta W_N = \frac{\partial \mathcal{L}}{\partial W_N}$ ,

$$\begin{aligned}\Delta W_N &= \frac{\partial \mathcal{L}}{\partial W_N} = \delta_N \\ \Delta W_{N-1} &= \delta_N w_{N-1} \frac{\partial \mathcal{L}}{\partial W_{N-1}} \\ \Delta W_{N-2} &= \delta_{N-1} w_{N-2} \frac{\partial \mathcal{L}}{\partial W_{N-2}} \\ &\vdots \\ \Delta W_0 &= \delta_1 w_0 \frac{\partial \mathcal{L}}{\partial W_0}\end{aligned}$$

We can write the general form for the weight updates  $\Delta W_{i \rightarrow j} = \delta_j z_i$ .

## Exercise-2

To solve this exercise we used the following python script.

```
import numpy as np

# weights from input to unit i
W = np.array([[0.60, 0.70, 0.00], [0.01, 0.43, 0.88]])
# weights from i to unit out
w = np.array([0.02, 0.03, 0.09])
# samples
x = np.array([[0.75, 0.8], [0.2, 0.05], [-0.75, 0.8], [0.2, -0.05]])
# target
y = np.array([1, 1, -1, -1])
# learning rate
theta = 0.5
```

We will use ReLU at every unit  $i$  and  $\tanh$  at the unit  $out$ .

```
def relu(x): return x*(x>0) # ReLU
def d_relu(x): return 1*(x>0) # Derivative of ReLU
def tanh(x): return np.tanh(x) # tanh
def d_tanh(x): return 1.0 - tanh(x)**2 # Derivative of tanh
```

Our loss function:

```
def error(x,y): return .5*(x-y)**2
```

And finally the main code, we perform weight updates every each sample, batch size is equal to one here.

```

for iteration in range(4):
    for i in range(4):
        _x = x[i] # sample
        _y = y[i] # target

        s_i = np.dot(_x, W) # input to unit s_i
        z_i = relu(s_i) # output of unit s_i

        s_out = np.dot(z_i, w) # input of units s_out
        z_out = tanh(s_out) # output of units s_out

        L = error(z_out, _y) # loss

        delta_out = (z_out - _y) * d_tanh(s_out) # Error signal at output unit out
        delta_i = delta_out * w.T * d_relu(s_i) # Error signal at unit i

        Delta_w = - theta * delta_out * z_i # Weight derivative at out
        Delta_W = - theta * delta_i * _x.reshape((2,1)) # Weight derivative at i

        w = w + Delta_w # weight updates
        W = W + Delta_W

```

Here we present the results for all iterations and all samples:

```

-Iteration: 0
---- sample: 0
  _x: [ 0.75  0.8 ]
  _y: 1
  s_i: [ 0.458  0.869  0.704]
  z_i: [ 0.458  0.869  0.704]
  s_out: 0.09859
  z_out: 0.0982718058711
  L: 0.406556868043
  delta_out: -0.893019891311
  delta_i: [-0.0178604 -0.0267906 -0.08037179]
  Delta_w: [ 0.20450156  0.38801714  0.314343 ]
  Delta_W: [[ 0.00669765  0.01004647  0.03013942]
[ 0.00714416  0.01071624  0.03214872]]
  w: [ 0.22450156  0.41801714  0.404343 ]
  W: [[ 0.60669765  0.71004647  0.03013942]
[ 0.01714416  0.44071624  0.91214872]]

---- sample: 1
  _x: [ 0.2  0.05]
  _y: 1
  s_i: [ 0.12219674  0.16404511  0.05163532]
  z_i: [ 0.12219674  0.16404511  0.05163532]
  s_out: 0.116885404762
  z_out: 0.11635593899
  L: 0.390413364759
  delta_out: -0.871680599695
  delta_i: [-0.19569365 -0.36437743 -0.35245795]
  Delta_w: [ 0.05325826  0.07149747  0.02250475]
  Delta_W: [[ 0.01956937  0.03643774  0.0352458 ]
[ 0.00489234  0.00910944  0.00881145]]
  w: [ 0.27775982  0.48951461  0.42684776]
  W: [[ 0.62626701  0.74648422  0.06538522]
[ 0.0220365  0.44982567  0.92096016]]

---- sample: 2
  _x: [-0.75  0.8 ]
  _y: -1
  s_i: [-0.45207106 -0.20000262  0.68772922]
  z_i: [-0. -0. 0.68772922]
  s_out: 0.29355673523
  z_out: 0.285404160873
  L: 0.826131928395
  delta_out: 1.1807008772
  delta_i: [ 0. 0. 0.50397952]
  Delta_w: [ 0. 0. -0.40600125]
  Delta_W: [[ 0. 0. 0.18899232]
[-0. -0. -0.20159181]]
  w: [ 0.27775982  0.48951461  0.02084651]
  W: [[ 0.62626701  0.74648422  0.25437754]
[ 0.0220365  0.44982567  0.71936836]]

---- sample: 3
  _x: [ 0.2 -0.05]
  _y: -1
  s_i: [ 0.12415158  0.12680556  0.01490709]
  z_i: [ 0.12415158  0.12680556  0.01490709]
  s_out: 0.0968682546849
  z_out: 0.0965664011817
  L: 0.6012289361
  delta_out: 1.08634084291
  delta_i: [ 0.30174183  0.53177972  0.02264641]
  Delta_w: [-0.06743546 -0.06887703 -0.00809709]
  Delta_W: [[-0.03017418 -0.05317797 -0.00226464]
[ 0.00754355  0.01329449  0.00056616]]
  w: [ 0.21032435  0.42063758  0.01274942]
  W: [[ 0.59609283  0.69330625  0.25211289]
[ 0.02958005  0.46312017  0.71993452]]

[ 0.07297015  0.54967053  0.72585295]]

---- sample: 2
  _x: [-0.75  0.8 ]
  _y: -1
  s_i: [-0.43383233 -0.16983587  0.37899681]
  z_i: [-0. -0. 0.37899681]
  s_out: 0.0820001176938
  z_out: 0.0818168205507
  L: 0.585163816613
  delta_out: 1.07457514727
  delta_i: [ 0. 0. 0.23249612]
  Delta_w: [ 0. 0. -0.20363028]
  Delta_W: [[ 0. 0. 0.08718605]
[-0. -0. -0.09299845]]
  w: [ 0.36644777  0.68936904  0.01273069]
  W: [[ 0.65627793  0.81276307  0.35610011]
[ 0.07297015  0.54967053  0.63285451]]

---- sample: 3
  _x: [ 0.2 -0.05]
  _y: -1
  s_i: [ 0.12760708  0.13506909  0.0395773 ]
  z_i: [ 0.12760708  0.13506909  0.0395773 ]
  s_out: 0.140377622452
  z_out: 0.139462745112
  L: 0.649187673749
  delta_out: 1.11730035735
  delta_i: [ 0.40943223  0.77023227  0.01422401]
  Delta_w: [-0.07128772 -0.07545637 -0.02210986]
  Delta_W: [[-0.04094322 -0.07702323 -0.0014224 ]
[ 0.01023581  0.01925581  0.0003556 ]]
  w: [ 0.29516006  0.61391267 -0.00937917]
  W: [[ 0.61533471  0.73573984  0.35467771]
[ 0.08320596  0.56892634  0.63321011]]

-Iteration: 2
---- sample: 0
  _x: [ 0.75  0.8 ]
  _y: 1
  s_i: [ 0.5280658  1.00694595  0.77257637]
  z_i: [ 0.5280658  1.00694595  0.77257637]
  s_out: 0.766794678598
  z_out: 0.645061746319
  L: 0.0629905819632
  delta_out: -0.207246793532
  delta_i: [-0.06117098 -0.12723143  0.0019438 ]
  Delta_w: [ 0.05471997  0.10434316  0.08005699]
  Delta_W: [[ 0.02293912  0.04771179 -0.00072893]
[ 0.02446839  0.05089257 -0.00077752]]
  w: [ 0.34988003  0.71825583  0.07067782]

```

```

W: [[ 0.63827382 0.78345163 0.35394879]
[ 0.10767435 0.61981891 0.63243258]]

---- sample: 1
_x: [ 0.2 0.05]
_y: 1
s_i: [ 0.13303848 0.18768127 0.10241139]
z_i: [ 0.13303848 0.18768127 0.10241139]
s_out: 0.188588887458
z_out: 0.18638447326
L: 0.330985112676
delta_out: -0.785351197118
delta_i: [-0.2747787 -0.56408307 -0.05550691]
Delta_w: [ 0.05224097 0.07369786 0.04021445]
Delta_W: [[ 0.02747787 0.05640831 0.00555069]
[ 0.00686947 0.01410208 0.00138767]]
w: [ 0.40212099 0.79195368 0.11089227]
W: [[ 0.66575169 0.83985993 0.35949948]
[ 0.11454381 0.63392099 0.63382026]]

---- sample: 2
_x: [-0.75 0.8 ]
_y: -1
s_i: [-0.40767872 -0.12275816 0.2374316 ]
z_i: [-0. -0. 0.2374316 ]
s_out: 0.0263293283635
z_out: 0.0263232459257
L: 0.526669702564
delta_out: 1.02561209292
delta_i: [ 0. 0. 0.11373245]
Delta_w: [ 0. 0. -0.12175636]
Delta_W: [[ 0. 0. 0.04264967]
[-0. -0. -0.04549298]]
w: [ 0.40212099 0.79195368 -0.01086409]
W: [[ 0.66575169 0.83985993 0.40214915]
[ 0.11454381 0.63392099 0.58832728]]

---- sample: 3
_x: [ 0.2 -0.05]
_y: -1
s_i: [ 0.12742315 0.13627594 0.05101347]
z_i: [ 0.12742315 0.13627594 0.05101347]
s_out: 0.158609538368
z_out: 0.157292741478
L: 0.669663244739
delta_out: 1.12866015421
delta_i: [ 0.45385794 0.89384657 -0.01226187]
Delta_w: [-0.07190871 -0.07690461 -0.02878843]
Delta_W: [[-0.04538579 -0.08938466 0.00122619]
[ 0.01134645 0.02234616 -0.00030655]]
w: [ 0.33021228 0.71504907 -0.03965252]
W: [[ 0.6203659 0.75047528 0.40337533]
[ 0.12589026 0.65626715 0.58802073]]

-Iteration: 3
---- sample: 0
_x: [ 0.75 0.8 ]
_y: 1
s_i: [ 0.56598664 1.08787018 0.77294808]
z_i: [ 0.56598664 1.08787018 0.77294808]
s_out: 0.934126957935
z_out: 0.732512223358
L: 0.0357748553266
delta_out: -0.123960748284
delta_i: [-0.04093336 -0.08863802 0.00491536]
Delta_w: [ 0.03508006 0.0674266 0.04790761]
Delta_W: [[ 0.01535001 0.03323926 -0.00184326]
[ 0.01637334 0.03545521 -0.00196614]]
w: [ 0.36529234 0.78247567 0.00825509]
W: [[ 0.63571591 0.78371453 0.40153207]
[ 0.14226361 0.69172236 0.58605459]]

---- sample: 1
_x: [ 0.2 0.05]
_y: 1
s_i: [ 0.13425636 0.19132902 0.10960914]
z_i: [ 0.13425636 0.19132902 0.10960914]
s_out: 0.199657961686
z_out: 0.197046584523
L: 0.322367093713
delta_out: -0.771776856983
delta_i: [-0.28192418 -0.60389662 -0.00637109]

Delta_w: [ 0.05180798 0.07383166 0.0422969 ]
Delta_W: [[ 0.02819242 0.06038966 0.00063711]
[ 0.0070481 0.01509742 0.00015928]]
w: [ 0.41710032 0.85630733 0.05055199]
W: [[ 0.66390833 0.8441042 0.40216918]
[ 0.14931171 0.70681978 0.58621386]]

---- sample: 2
_x: [-0.75 0.8 ]
_y: -1
s_i: [-0.37848188 -0.06762233 0.1673442 ]
z_i: [-0. -0. 0.1673442 ]
s_out: 0.00845958221704
z_out: 0.00845938042081
L: 0.508495160979
delta_out: 1.00838721394
delta_i: [ 0. 0. 0.05097598]
Delta_w: [ 0. 0. -0.08437388]
Delta_W: [[ 0. 0. 0.01911599]
[-0. -0. -0.02039039]]
w: [ 0.41710032 0.85630733 -0.03382189]
W: [[ 0.66390833 0.8441042 0.42128518]
[ 0.14931171 0.70681978 0.56582347]]

---- sample: 3
_x: [ 0.2 -0.05]
_y: -1
s_i: [ 0.12531608 0.13347985 0.05596586]
z_i: [ 0.12531608 0.13347985 0.05596586]
s_out: 0.164676279904
z_out: 0.163203672672
L: 0.676521392059
delta_out: 1.13222123247
delta_i: [ 0.47224984 0.96952934 -0.03829386]
Delta_w: [-0.07094276 -0.07556436 -0.03168287]
Delta_W: [[-0.04722498 -0.09695293 0.00382939]
[ 0.01180625 0.02423823 -0.00095735]]
w: [ 0.34615756 0.78074297 -0.06550476]
W: [[ 0.61668334 0.74715126 0.42511456]
[ 0.16111796 0.73105801 0.56486613]]

```

## Exercise-3

i) Since  $\max(0, p_j - p_{y_i} + \text{margin})$  is minimized when  $p_j - p_{y_i} = -\text{margin}$  which results  $p_j = p_{y_i} - \text{margin}$ , the loss function is trying to maximize the difference of the probability output for the class  $p_{y_i}$  with regards to every other class  $j$  by the value of  $\text{margin}$ . In simple words,  $\mathcal{L}_{\text{hinge}}$  is trying to maximize the probability difference between the correct and all other classes.

ii)

$$\frac{\partial \mathcal{L}_{\text{hinge}}}{\partial o_j} = \frac{\partial}{\partial o_j} (\max(0, p_j - p_{y_i} + \text{margin}))$$

For  $p_{y_i} > p_j + \text{margin}$ ,  $\frac{\partial \mathcal{L}_{hinge}}{\partial o_j} = 0$ , while for  $p_{y_i} = p_j + \text{margin}$ ,  $\frac{\partial \mathcal{L}_{hinge}}{\partial o_j} = \emptyset$ . We assume  $p_{y_i} < p_j + \text{margin}$ ,

$$\begin{aligned}
\frac{\partial \mathcal{L}_{hinge}}{\partial o_j} &= \frac{\partial}{\partial o_j} (p_j - p_{y_i} + \text{margin}) \\
&= \frac{\partial}{\partial o_j} p_j \\
&= \frac{\partial}{\partial o_j} \left( \frac{\exp(o_j)}{\sum_k \exp(o_k)} \right) \\
&= \frac{\partial}{\partial o_j} \left( \exp(o_j) \frac{1}{\sum_k \exp(o_k)} \right) \\
&= (\exp(o_j))' \frac{1}{\sum_k \exp(o_k)} + \exp(o_j) \left( \frac{1}{\sum_k \exp(o_k)} \right)' \\
&= \exp(o_j) \frac{1}{\sum_k \exp(o_k)} - \exp(o_j) \left( \frac{1}{\sum_k \exp(o_k)} \right)^2 \left( \sum_k \exp(o_k) \right)' \\
&= \exp(o_j) \frac{1}{\sum_k \exp(o_k)} - \exp(2o_j) \left( \frac{1}{\sum_k \exp(o_k)} \right)^2 \\
&= p_j - p_j^2
\end{aligned}$$

which is the derivative of the loss function  $\mathcal{L}_{hinge}$  with respect to  $o_j$ .