COMP9444
Neural Networks and Deep Learning
Term 2, 2025

Week 2 Tutorial Solutions

This page was last updated: 06/14/2025 13:35:45

1. **Do you have any questions from the first tutorial? Discuss them with your tutor.**

   Note: Sample Solutions for the Week 1 Tutorial are released on WebCMS. Please read the sample solutions and check that you understand them in detail.

2. **Simple Gradient Descent by Hand**

   Consider the simplest possible machine learning task:

   Solve $f(x) = wx$ such that $f(1) = 1$, i.e. $f(x) = t$, for $x = 1$, $t = 1$.

   We can solve this by gradient descent using the loss function

   $E = ½(wx - t)^2$, with learning rate of $\eta = 0.5$ and initial value $w = 0$.

   a. Perform the first epoch of training by completing this table:

   $w = 0$
   $x = 1$
   $f(x) = ?$
   $E = ?$
   $\partial E/\partial w = ?$
   $w \leftarrow w - ? = ?$

   $w = 0$
   $x = 1$
   $f(x) = wx = (0)(1) = 0$
   $E = ½(wx - 1)^2 = ½(0\times1 - 1)^2 = 0.5$
   $\partial E/\partial w = x(wx - 1) = 1(0\times1 - 1) = -1$
   $w \leftarrow w - \eta\ \partial E/\partial w = 0 - 0.5(-1) = 0.5$

   ---

   b. Repeat these calculations for the second epoch.

   $w = 0.5$
   $x = 1$
   $f(x) = wx = (0.5)(1) = 0.5$

$E = \frac{1}{2}(wx - 1)^2 = \frac{1}{2}(0.5\times1 - 1)^2 = \frac{1}{2}(-0.5)^2 = 0.125$

$\partial E/\partial w = x(wx - 1) = 1(0.5\times1 - 1) = -0.5$

$w \leftarrow w - \eta\, \partial E/\partial w = 0.5 - 0.5(-0.5) = 0.75$

---

c. Copy the following code into a file called `slope.py`, and use it to explore what happens for different values of the learning rate.
Keeping the momentum fixed at 0, try each of the values for `lr` listed below and describe what happens for each value, in terms of the success and speed of the algorithm.
`lr = 0.01, 0.1, 0.5, 1.0, 1.5, 1.9, 2.0, 2.1`
Discuss your findings in class.

```
import torch
import torch.utils.data
import numpy as np

lr = 1.9 # learning rate
mom = 0.0 # momentum

class MyModel(torch.nn.Module):
    def __init__(self):
        super(MyModel, self).__init__()
        self.w = torch.nn.Parameter(torch.zeros((1),
                                    requires_grad=True))
    def forward(self, input):
        output  = self.w * input
        return(output)

device = 'cpu'

input  = torch.Tensor([[1]])
target = torch.Tensor([[1]])

slope_dataset = torch.utils.data.TensorDataset(input,target)
train_loader  = torch.utils.data.DataLoader(slope_dataset,batch_size=1)

# create neural network according to model specification
net = MyModel().to(device) # CPU or GPU

# choose between SGD, Adam or other optimizer
optimizer=torch.optim.SGD(net.parameters(),lr=lr,momentum=mom)

epochs = 1000
```

```
epoch = 0
while epoch < epochs:
    epoch = epoch + 1
    for batch_id, (data,target) in enumerate(train_loader):
        optimizer.zero_grad() # zero the gradients
        output = net(data)    # apply network
        loss = 0.5*torch.mean((output-target)*(output-target))
        print('Ep%3d: zero_grad(): w.data=%7.4f loss=%7.4f' \
                     % (epoch, net.w.data, loss))
        loss.backward()       # compute gradients
        optimizer.step()      # update weights
        print('            step(): w.grad=%7.4f w.data=%7.4f' \
                     % (net.w.grad, net.w.data))
        if loss < 0.000000001 or np.isnan(loss.data.item()):
            epoch = epochs
            break
```

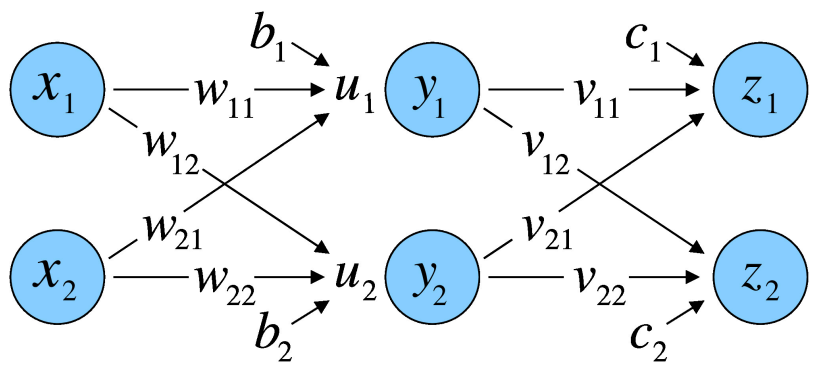
```
0.01  the task is learned in 998 epochs
0.1   the task is learned in  97 epochs
0.5   the task is learned in  16 epochs
1.0   the task is learned in   2 epoch
1.5   the task is learned in  16 epochs
1.9   the task is learned in  97 epochs
2.0   the parameter oscillates between 0.0 and 2.0
2.1   the parameter explodes, to inf and then to nan
```

d. **Bonus Exercise**: If you have learned about momentum, try running the above code with different values of `momentum` between `0.1` and `0.9`, while keeping the learning rate fixed at `1.9`. For which value of momentum is the task solved in the fewest epochs? What happens when the momentum is `1.0`? And when it is `1.1`?

The fewest is 13 epochs, when momentum is 0.3.

When momentum is 1.0, the parameter cycles around and never converges. When momentum is 1.1, the parameter blows up to Infinity.

3. **[illegible]anding Backpropagation**



Consider a 2-layer neural network as shown above, with ReLU activations at the hidden nodes (i.e. $y_1 = \mathrm{ReLU}(u_1) = \{u_1, \text{if } u_1 > 0;\ \ 0, \text{otherwise}\}$). There is no activation function at the output nodes. Suppose the initial weights are:

$$w_{11} = 0.5,\ w_{12} = –0.7,\ w_{21} = 0.4,\ w_{22} = –0.6,\quad b_1 = –0.9,\ b_2 = 0.7$$
$$v_{11} = 0.3,\quad v_{12} = –0.6,\ v_{21} = –0.4,\ v_{22} = –0.8,\quad c_1 = –0.3,\ c_2 = 0.2$$

Let the training set consist of just one item, with input (x_1, x_2) = (0.5, –0.4) and target output (t_1, t_2) = (1.0, 0.0).

a. Do a forward pass through the network to calculate by hand the values of: $u_1, u_2, y_1, y_2, z_1, z_2$

$$u_1 = b_1 + x_1w_{11} + x_2w_{21} = –0.9 + (0.5)(0.5) \quad + (–0.4)(0.4) = –0.81$$
$$u_2 = b_2 + x_1w_{12} + x_2w_{22} = \quad 0.7 + (0.5)(–0.7) + (–0.4)(–0.6) = 0.59$$
$$y_1 = \mathrm{ReLU}(u_1) = 0$$
$$y_2 = \mathrm{ReLU}(u_2) = 0.59$$
$$z_1 = c_1 + y_1v_{11} + y_2v_{21} = –0.3 + (0)(0.3) \quad + (0.59)(–0.4) = –0.536$$
$$z_2 = c_2 + y_1v_{12} + y_2v_{22} = \quad 0.2 + (0)(–0.6) + (0.59)(–0.8) = –0.272$$

b. Calculate the loss function: $\mathrm{E} = ½\Sigma_i(z_i – t_i)^2$

$$\mathrm{E} = ½[(z_1 – t_1)^2 + (z_2 – t_2)^2] = ½[(–0.536 – 1.0)^2 + (–0.272 – 0.0)^2] = 1.21664$$

c. Use backpropagation to calculate the values of the differentials:
$dz_1, dz_2, dc_1, dc_2, dv_{11}, dv_{12}, dv_{21}, dv_{22}, dy_1, dy_2,$
$du_1, du_2, db_1, db_2, dw_{11}, dw_{12}, dw_{21}, dw_{22}$ (where $dz_1 = \partial\mathrm{E}/\partial z_1$, etc.)

$$dz_1 = \partial\mathrm{E}/\partial z_1 = (z_1 - t_1) = (–0.536 – 1.0) = –1.536$$
$$dz_2 = \partial\mathrm{E}/\partial z_2 = (z_2 - t_2) = (–0.272 – 0.0) = –0.272$$

$dc_1 = \partial z_1/\partial c_1 \times \partial E/\partial z_1 = 1\times dz_1 = -1.536, \quad dc_2 = dz_2 = -0.272$

$dv_{11} = \partial z_1/\partial v_{11} \times \partial E/\partial z_1 = y_1 dz_1 = 0, \quad dv_{12} = y_1 dz_2 = 0$

$dv_{21} = \partial z_1/\partial v_{21} \times \partial E/\partial z_1 = y_2 dz_1 = (0.59)(-1.536) = -0.90624$

$dv_{22} = \partial z_2/\partial v_{22} \times \partial E/\partial z_2 = y_2 dz_2 = (0.59)(-0.272) = -0.16048$

$dy_1 = \partial z_1/\partial y_1 \times \partial E/\partial z_1 + \partial z_2/\partial y_1 \times \partial E/\partial z_2$

$\quad = v_{11} dz_1 + v_{12} dz_2 = (0.3)(-1.536) + (-0.6)(-0.272) = -0.2976$

$dy_2 = v_{21} dz_1 + v_{22} dz_2 = (-0.4)(-1.536) + (-0.8)(-0.272) = 0.832$

$du_1 = \partial y_1/\partial u_1 \times \partial E/\partial y_1 = H(u_1) dy_1 = 0, \quad du_2 = H(u_2) dy_2 = 0.832$

$db_1 = \partial u_1/\partial b_1 \times \partial E/\partial u_1 = 1\times du_1 = 0, \quad db_2 = du_2 = 0.832$

$dw_{11} = x_1 du_1 = (0.5)(0) = 0, \; dw_{12} = x_1 du_2 = (0.5)(0.832) = 0.416$

$dw_{21} = x_2 du_1 = (-0.4)(0) = 0, \; dw_{22} = x_2 du_2 = (-0.4)(0.832) = -0.3328$

d. With learning rate $\eta = 0.5$, calculate the updated weight values: $w_{11}, w_{12}, w_{21}, w_{22}, b_1, b_2, v_{11}, v_{12}, v_{21}, v_{22}, c_1, c_2$

$w_{11} \leftarrow w_{11} - \eta dw_{11} = 0.5 - 0.5(0) = 0.5$

$w_{12} \leftarrow w_{12} - \eta dw_{12} = -0.7 - 0.5(0.416) = -0.908$

$w_{21} \leftarrow w_{21} - \eta dw_{21} = 0.4 - 0.5(0) = 0.4$

$w_{22} \leftarrow w_{22} - \eta dw_{22} = -0.6 - 0.5(-0.3328) = -0.4336$

$b_1 \leftarrow b_1 - \eta db_1 = -0.9 - 0.5(0) = -0.9$

$b_2 \leftarrow b_2 - \eta db_2 = 0.7 - 0.5(0.832) = 0.284$

$v_{11} \leftarrow v_{11} - \eta dv_{11} = 0.3 - 0.5(0) = 0.3$

$v_{12} \leftarrow v_{12} - \eta dv_{12} = -0.6 - 0.5(0) = -0.6$

$v_{21} \leftarrow v_{21} - \eta dv_{21} = -0.4 - 0.5(-0.90624) = 0.05312$

$v_{22} \leftarrow v_{22} - \eta dv_{22} = -0.8 - 0.5(-0.16048) = -0.71976$

$c_1 \leftarrow c_1 - \eta dc_1 = -0.3 - 0.5(-1.536) = 0.468$

$c_2 \leftarrow c_2 - \eta dc_2 = 0.2 - 0.5(-0.272) = 0.336$

4. **Paper Discussion**

The aim of paper discussion is to analyse the research paper from a variety of different perspectives. During the second half of your tutorial, your tutor will randomly divide up your class into panels (or groups), each panel having 4 or 5 students with each student having a defined role. You need to critically read the paper from that particular perspective. Each member in the panel should [illegible] te by sharing their thoughts or asking questions. These paper [illegible] ons will be moderated by your tutors in the actual tutorial session. [illegible] r, we highly encourage you to read the paper before coming to the class.

It's OK if you don't understand each aspect of technical details but you need a get a high-level idea of the paper. These paper discussions and roles defined below are taken from or inspired by Colin Raffel and Alec Jacobsen's role-playing paper-reading seminars.

- **Archaeologist**: You are an archaeologist who must determine where this paper sits in the context of previous work. Find one **older** paper cited within the current paper that substantially influenced the current paper and be prepared to discuss what is new. Trace each aspect of the paper (e.g., model, training, data) back to prior work.
- **Social Impact Assessor**: You are an auditor of the societal impact of the paper. Identify how this paper self-assesses its real-world impact (both positive and negative). What are possible negative social impacts that were overlooked or omitted?
- **Industry Practitioner**: You work at a company or organisation developing an application or product of your choice (that has not already been suggested in a prior session). Bring a convincing pitch for why you should be paid to implement the method in the paper and discuss at least one positive and negative impact of this application.
- **Researcher**: You are a researcher who is working on a new project in this area. Propose an imaginary follow-up project not just based on the current but only possible due to the existence and success of the current paper.
- **Scientific Peer Reviewer**: The paper has not been published yet and is currently submitted to a top conference where you've been assigned as a peer reviewer. Check various aspects such as reproducibility, rigor, correctness, clarity, novelty and comment on whether to accept or reject the paper.

Time and again, humanity has demonstrated that when we work together and apply our collective mind, we can forge solutions to seemingly intractable problems. You can also read these papers from United Nations Sustainable Development Goals (SDGs) viewpoint. Recently, the UN has released UN Sustainable Development Goals Report 2024 which you can read to track how we are progressing towards SDGs (optional). In Week 2, we will be reading the following paper:

- Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks, NeurIPS, 2012.

2 ıe AlexNet paper, one of the seminal papers which sparked an "Aha t" in the research community by a 10% reduction in error rate on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 and

sparked the deep learning revolution. You can download paper by clicking on the paper title or by copy-pasting this URL:
`https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`

The Week 2 checkpoint is to participate in the paper reading session in the tutorial session.

5. **Any Other Questions**

   Any further questions or discussion about the Week 1 material, other parts of the course, or broader implications of deep learning.

2