Exercises INF 5860

Solution hints

Please note that the solution hints are in terms of keywords, and during an exam you might elobarate some more.

Exercise 1 Linear regression

a) What is the loss function for linear regression?

The squared error $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$

b) Why would we want an iterative algorithm for the linear regression problem?

Although an analytical solution exist, with an iterative solution we don't need to invert X^TX which might be non-invertible for large data sets.

c) How does gradient descent update the estimate, give the general formulae?

$$\theta^j = \theta^j - \varepsilon \frac{\partial}{\partial \theta^j} J(\theta^0 \ , \theta^1)$$

d) Given x= $x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad y = \begin{bmatrix} 1.5 \\ 2 \\ 2.5 \end{bmatrix}$

Plot x,y as points in a plot.

e) If we start with Θ_0=0 and Θ_1=0, compute the value of the initial loss function

J= 1/6(1.5*1.5+2*2+2.5*2.5)=12.5/6=2.08

f) If we start with Θ_0=0 and Θ_1=0, compute the estimate after one iteration if the learning rate is 1.

$$\frac{\partial}{\partial w} J(w,b) = \frac{\partial}{\partial w} \frac{1}{2m} \sum_i \left(w \; x_i + b - y_i\right)^2$$

$$= \frac{2}{2m} \sum_i \left(w \; x_i + b - y_i\right) x_i = 1/3 * (1*1.5 + 2*2 + 2.5*3) = 13/3 = 4.33$$

$$\frac{\partial}{\partial b} J(w,b) = \frac{\partial}{\partial b} \frac{1}{2m} \sum_i \left(w \; x_i + b - y_i\right)^2$$

$$= \frac{2}{2m} \sum_i \left(w \; x_i + b - y_i\right) = 1/3 * (1.5 + 2 + 2.5) = 6/3 = 2$$

So with learning rate 1, and Θ_0=0 and Θ_1=0, after 1 iteration the estimates are W=-4.33 and b=-2.

Exercise 2 – Logistic classification

a) Given a trained logistic classifier for a single feature and 2 classes. What is the equation for the decision boundary if W=2 and b=1?

The equation for the boundary is θ^Tx=0, or Wx+b=0, which gives 2x+1=0, or x=-1/2.

b) With logistic classification, how is a new sample classified?

A new sample is classified to class 1 if h_Θ(x)>0, to class 0 otherwise.

c) How can we generalize logistic classification to more than 2 classes?

With a logistic classifier, train one classifier per class and select the class c that maximize h_{Θ_C}(x)

Exercise 3: Basic neural networks

a) What are the problems with using a sigmoid activation function?

Basically two main problems, it kill gradients, and is not zero-centered.

b) In what way does the tanh activation function share the same drawbacks?

It is zero-centered, but still kills gradients.

c) Discuss briefly why feature scaling of the input features is important.

Fill this in yourself…

d) Why is initializing all the weights to zero problematic?

If all weights are equal, nodes will learn the same thing during backpropagation, and this limits the capacity.

e) Assume that we have a 2-layer net (one hidden layer) with weights $W^{(1)}$, $b^{(1)}$, $W^{(2)}$ and $b^{(2)}$. Assume that we use RELU-activations in the hidden layer, and no activation on the output layer. Write down an equation for the output of the j'th node in the hidden layer, $a^{(j)}$.

$a^{(j)}=max(W^{(1)}X+ b^{(1)},0)$

f) Explain shortly how maxnorm regularization works.

Maxnorm sets a limit to the maximum sum of the input weights to a node. If it is larger, it will be contrained down to the selected maximum value.

g) When using dropout, if you do not consider any scaling during training, how should you then compensate during prediction of new data/test samples?

If scaling is not done during training, each output in layers affected by dropout must be scaled by p, the dropout probability.

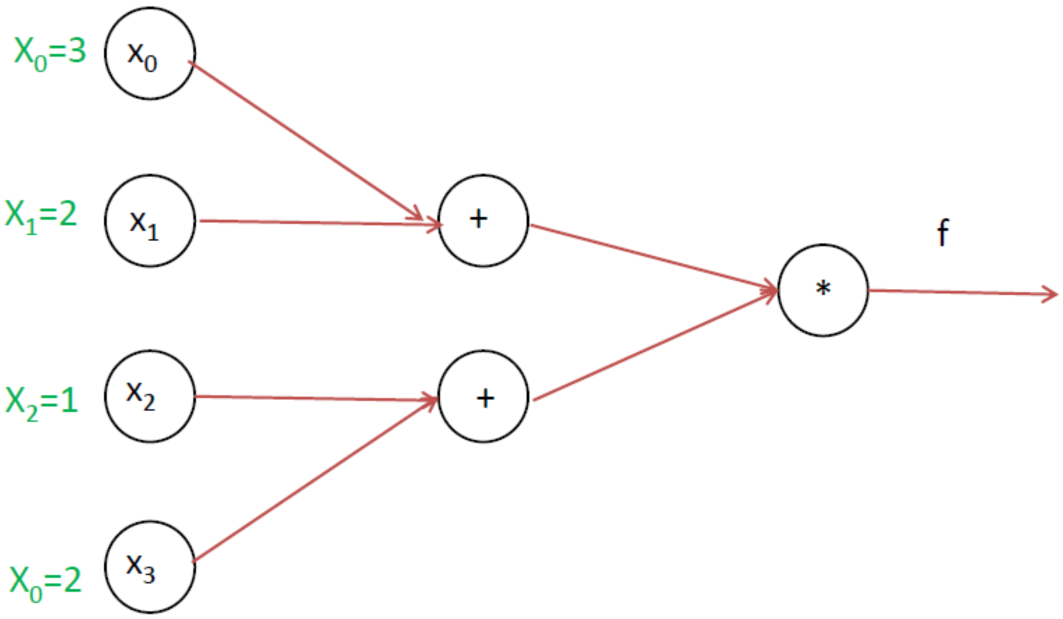
h) Explain briefly how momentum gradient descent works, and why this can be more robust than regular gradient descent.

```
v=mu*v - learning_rate*df  # Integrate velocity
f += v                     #Integrate position
```

This acts like friction, allows velocity to build up in shallow directions, but is dampened in steep directions because of the sign change.

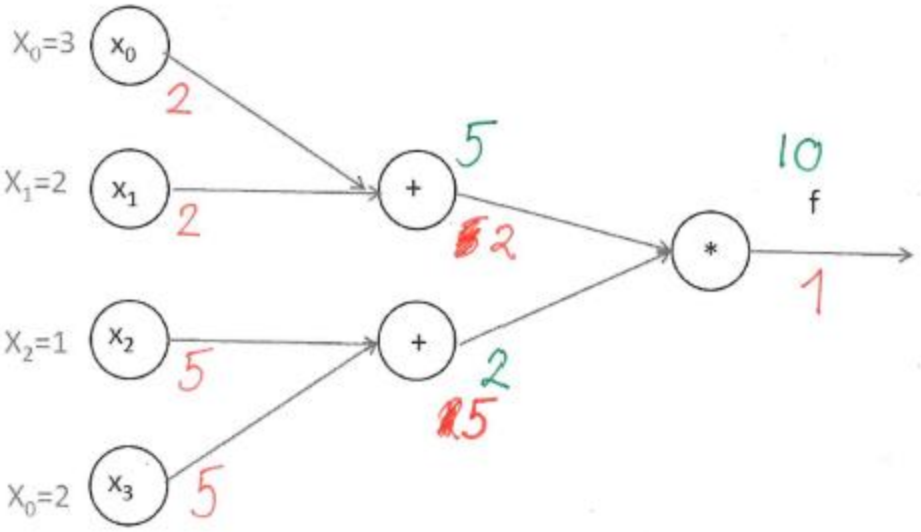
Exercise 3: A simple network

5



a) Perform backpropagation on this graph

Backpropagated values in red.



Exercise 4: Generalization

1. Why can testing out multiple models on your test data be a problem and when is it problematic?

   Your test data is intended to use for the final evaluation of your model. Choosing the model that works best can make the model overestimate the result. It is problematic when you make decisions based on the test data, e.g. choosing a model. You can of course test many models and report results of all models.

2. How does searching through more hypotheses affect the probability of searching through a solution close to the correct solution?

   You are more likely to search through a solution close to the correct.

3. Give an example of a way to measure model complexity.

   Vapnik–Chervonenkis (VC) dimension is perhaps the most common measure of complexity. It measures the highest number of points to which a given model can perfectly fit any binary labeling of the points.

4. What are the implications of the "No free lunch" theorem mean for machine learning?

   No free lunch theorem states that no classifier are better than any other for all possible distributions. Very briefly, this means that you always need some assumptions about the application for the model, to make learning possible.

5. Give three examples of common assumptions (priors) machine learning models make.
   Local consistency, stating that points that are close in space usually have similar labels.
   Shared factors, different classes share some representations
   Linearity, relationships between variables are linear

Exercise 5: Representations

1. What do we mean when we refer to the image manifold?

   We mean the connected space where all values in that space can be said to belong to a given class or group.

2. Explain why working with image gradients can be better than working with raw pixels. What additional effect can you achieve by scaling the gradients based on the gradient magnitude?

   Image gradients are invariant to changes in brightness (added or subtracted values). If changes in brightness are irrelevant for the application, using image gradients may require *less samples, as you don't need to learn that images can have different brightness. If you also scale the images using gradient magnitude your model will also be invariant to contrast changes.*

3. How can multiple layers of discriminators/classifiers reduce the need for training examples in image analysis?

   If examples from different classes share representations/factors, a model with multiple layers can use examples from all classes to train/fit those shared representations. The final layer may therefore be able to learn a much simpler function, that requires less samples.

Exercise 6: Convolutional nets

1. You have a 32x32x5 image and filter it with a 5x5x5 kernel, the way most convolutional neural networks are implemented. If you use no padding, what will be the output size of the activation map?

   28x28x1

2. What do we mean by dilated convolutions and how are they used?

   In dilated convolutions, we fill a given factor of the weights with zero, or just skip them entirely. With this approach, we can get a larger field-of-view with less weights. It is typical to increase the dilation factor throughout the network, so we can maximize the field-of-view, but still not miss any parts of the image.

3. Why is the effective field-of-view usually smaller than the theoretical field-of-view? *By theoretical field-of-view we mean the size of the image patch that can influence each of the output values in the activation map. Practical field-of-view is the size of the patch of pixels influencing the results of a given output value.*

   The values in the outer fringes of the field-of-view are only included with one chain of multiplications, while the values near the center (spatially similar position as the output value), influence the results through many paths. The values included only a few times, are more likely to be drowned by noise, and in practice not influencing the result at all, due to rounding errors, or simply not passing through ReLus.

4. In deep learning frameworks, you usually operate on 4D tensors, when working with 2D convolutions. If you want to use such a framework to do an average (blur) filtering of images, how would you have to construct the kernel for the convolution? *You should treat each of the color channels (RGB) independently.*

   We can view the two non-spatial dimensions as a matrix multiplication, where each cross-term indicates the interactions between the channels. In other words the non-spatial dimensions should be identity matrices, or identity matrices divided by the spatial size. We can use the same identity matrix at every spatial position.

Exercise 7: Training deep networks

1. Gradient flow
    a. Why is *gradient flow* important when training deep neural networks?
    b. Give some common methods that help to ensure good gradient flow.

    a) The gradients have information on how the network should be updated to perform better. If the gradients are disturbed and does not reach all the weights, those weights are not updated. It may of course be that those weights should not be updated, but often the gradients are lost from rounding errors and other problems.

    b) Common methods are: careful initialization, batch normalization, leaky Relu's, ResNet.

2. How does batch size relate to learning rate? Explain.

    A small batch size can give similar effects as high learning rate, and vice versa. With a large batch size, the gradient from a given example is averaged with the contributions from many other examples. Therefore, the relative contribution, or gradient from a given example is relatively small. Similarly, a small learning rate also make the contribution from one single example becomes smaller.

3. Why is it a problem to optimize accuracy directly with a deep neural network?

    Accuracy with respect to the weights of a deep network resembles a step function. A small change in the weights will not necessarily change the accuracy at all, or it may change it a lot. The gradient of the accuracy function is in other words either zero, or not defined, which is obviously a problem when using gradient decent.

Exercise 8: Deep learning architectures

1. Give two possible explanations to why residual networks work better than standard feed forward networks.

    If the weights for a convolution in a residual layer is zero or close to zero, it won't stop some non-zero gradient from reaching prior layers. This can give better gradient flow and more possible weights to optimize.

    Residual can also much simpler keep the calculations at previous layers, as it just has to set the weight to zero to keep the features intact. This means that a network more easily can reuse representations, and may need less examples for training.

2. You want to find bounding-boxes for cars in an image. You don't know how many cars there will be in each image, but you can safely assume it's between 0 – 100. Describe how you can construct and train a deep neural network for this task.

A possible approach would be to build a network that outputs 5 values for each spatial location. The first 4 values represent a bounding-box, while the last value represent a confidence score. You need annotations of bounding-boxes for all the training images.

The first 4 values can be trained by regression loss, for the distances to the closest bounding-box. It may be a good idea to only train spatial positions close to actual bounding-boxes. The confidence value can be trained by softmax cross-entropy loss, with a target being true if the proposed bounding-box overlap by at least a given fraction, and false otherwise.

3. What does it mean to use a "Fully-convolutional" architecture for image segmentation?
   A fully-convolutional architecture means that you only use convolutions In your network and no-fully-connected layers. This means that you can input arbitrary sized images. Additionally it means that you can segment a whole image with a single run through the convolutional network. Since many of the positions throughout the network are used in many outputs, this means that you can save a lot of computation.

4. What is the reasoning behind the concatenation operations in U-Net for image segmentation?

   By concatenating the last layers, with layers before you reduced the spatial size, means that you can keep spatial information. This also means that the mid layers of the network don't have to keep spatial information, which means it can be more memory efficient.

Exercise 9: Visualization and Adversarial training

1. You have a convolutional neural network trained for image classification. Describe a simple way of detecting what parts of an image are responsible for a certain classification result, without using the image gradients.

   You can use occlusion. Set the values in a certain location to zero before you run inference on the network, and compare the results with the network run on the full image. Doing this, multiple times, and occluding different part of the image and comparing the softmax output of a given class, you find the important regions as the regions that reduce the softmax output by a lot.

2. How can you get a simple estimate of how changing a set of pixel-values will affect the final class probabilities?

   The gradients of the class probabilities with respect to the image are estimates of how the values in the image affect the output.

3. For some visualization techniques, you apply a lowpass (blurring) filter between each iteration of optimization. Why may this be a reasonable approach?

   This is done with the reasoning that natural images (real images) don't contain as much high frequency information as the random images or the image gradients. Making the image look *more similar to the images in the dataset will also make it easier to see what kind if images have what effect on the network.*

4. You have lots of training images for one application, but no labelled images for a similar application. How can you use Adversarial domain adaption, to improve your results on the new data?

Exercise 10: Recurrent Neural networks (RNN)

1. Why is vanishing gradients and outputs a more common problem in basic RNNs compared to feed forward networks?

   Since RNNs use the same weights for each iteration, it will also have the same effect on the input every time. If the effect is increasing the values, the values can easily become very big and if the effect is decreasing the values, they become small. RNNs often solve exploding gradients by using tanh activation functions and gradient clipping. Both are known for creating vanishing gradients.

2. Why is vanishing gradients and outputs in RNN less problematic than for feed forward neural networks?

   RNNs usually get input and give output for each step. This means that for only one time-step, it is similar to a one or two-layer network, and here we know that vanishing gradients are less of a problem. Also since an RNN reuse the same weights, they only need one of the iterations to have gradients for the weights at all iterations to get updated. This means that it is less likely that an RNN get totally stuck, due to these vanishing gradients.

3. Why can you only do gradient descent for a certain number iterations of an RNN and when is this a problem? Explain and provide an example.

   The question should be, why can you only do backprop for a certain number of iterations of an RNN at one time. To do backpropagation of a network, you need to keep the output values of each layer, until you get to the backpropagation step. This basically means that the number of iterations you can optimize jointly, is restricted by the computer memory. This is a problem if you want to learn interactions over many iterations. If you want to learn a relationship between an input and an output, you need to optimize jointly the iteration that takes in the output, and the iterations that give the output.

   A typical example could be that you are trying to predict the next word in a text. Whether you should use he or she in a sentence, may depend on a name given many pages earlier.

4. Give an overview of some common solutions to using deep learning for video data.

   The simplest way to use time information in video, may be to simply input video frames as channels in a CNN. In this setting you will only have a very limited "field-of-view" in the time domain. Additionally, you have not built into your network any knowledge regarding the *order of the frames. So, the network must learn how the order affect the results. An alternative is to use 3D convolutions, so you can have a convolutional network in the time-dimension as well. This build in the order of the frames, into the model, but there is no information of what is "forward", and what is "backward". This will also restrict somewhat the field of view.*

Another alternative is to first use a convolutional network, and then use an RNN instead of a fully-connected layer. This both build in the order, and that the first images will affect the next. You also get a theoretically infinite field of view in the time dimension, in test time.

A final alternative is to build each layer in the convolutional network into a recurrent layer.

Exercise 11: Reinforcement learning

1. Is Reinforcement learning usually training faster or slower than standard supervised learning?

   Reinforcement learning converge slower, as the gradients are more rare and less accurate.

2. In what kind of situations is it common to use Reinforcement learning? Explain why.

   In situations where you have non-differential functions you are trying to learn or the functions are hard to optimize by gradient decent alone. In reinforcement, instead of you don't need the direct gradient information, instead you simply update your model to make all actions leading up to a reward more likely and actions leading to punishment less likely.

3. In what type of situation does Policy learning require a lot of memory?

   Policy learning need to get at least one feedback (reward/punishment) to do an update step. Every step must keep its output, until it can be updated, so if you need many steps before an update, you will require lots of memory.

4. How could you implement hard attention for image analysis in a fully supervised way, without using reinforcement learning?

   Use R-CNN, as described in 8.2, only you input the cropped image into a new network.

Exercise 12: Unsupervised learning

1. Draw and explain an example where t-SNE work better than PCA.

   *If the samples are primarily similar to close neighbors, and the large distances between samples are less important, t-SNE can work better than PCA. A typical example is the **spiral**, where long distance is "irrelevant" and neighbor connectivity is important.*

2. When you do a PCA of a dataset, you can easily transform new points with the same transform. Why is it more difficult to transform new points with t-SNE?

   In t-SNE you work on the points directly, so if a point does not exist in the dataset it is not obvious, where to place it in the lower dimension. Some sort of interpolation may work, but not necessarily.

3. Give basic overview of what an autencoder based on neural networks is.

   An autoencoder try to replicate its input. To make it useful we put some restriction on a temporary representation of the data, that we often call embedding or latent representation. This means the an autoencoder learn to represent the given data in some alternate or compressed way.
   When it is based on a neural network, it simply means that it is a neural network that encodes the data into the embedding and decodes it back to its original space.

4. Explain a typical situation where first learning an embedding unsupervised and then using the embedding for supervised learning, can fail.

   An autoencoder typically optimize the l2 loss between the original input and the reconstruction. For images, this means that it will prioritize general color and position. Specific details will often not be part of the embedding. If your later application rely on those specific details, you may be in trouble and the supervised learning will fail, as you don’t have the information you need.