Linear Models

For a binary classification problem with a linear model, one simple idea is to consider 0-1 loss function to measure classifier's performance for a pair (x,y):

$$l(sgn(x^T w + b), y) = 1[sgn(x^T w + b) \neq y]$$

where sgn and 1 denote sign and indicator functions, respectively.

What is the main issue with this loss function?

A. This loss counts mispredictions.
B. There should not be a bias term.
C. This loss is non-differentiable with uninformative derivatives.
D. There is no l_2 regularization term to control the Euclidean norm of w.

Linear Models

For a binary classification problem with a linear model, one simple idea is to consider 0-1 loss function to measure classifier's performance for a pair (x,y):

$$l(sgn(x^T w + b), y) = 1[sgn(x^T w + b) \neq y]$$

where sgn and 1 denote sign and indicator functions, respectively.

What is the main issue with this loss function?

E. This loss counts mispredictions.
F. There should not be a bias term.
G. This loss is non-differentiable with uninformative derivatives.
H. There is no l_2 regularization term to control the Euclidean norm of w.

Solution: C

This loss counts mispredictions. Having a bias term is not an issue. Adding weight decay or other forms of regularization does not resolve the issue that this 0-1 loss is non-differentiable with uninformative derivatives.

Introduction to Neural Networks

Suppose you want to predict how many minutes per day your cat may ignore you. You consider these input features: 1) number of treats given x_1; 2) time spent petting x_2; 3) presence of a dog at home x_3; 4) presence of a cardboard box at home x_4; and 5) type of the food your cat has x_5. Let y denote the true output value and $\hat{y}$ denote the prediction. Suppose you use a neural network that predicts $\hat{y}$ as follows

$$\hat{y} = \sum_{j=1}^{7} v_j \phi\left(\sum_{i=1}^{5} w_{i,j} x_i + b_j\right) + b$$

where v_j, $w_{i,j}$, b_j, b are trainable parameters and ϕ is an activation function

Let L denote the number of layers, n_{hidden} denote the number of hidden nodes (neurons in hidden layers), and z_2 denote the output (activation) of neuron 2 in hidden layer 1. What are L, n_{hidden}, z_2?

A. $L = 1,\ n_{hidden} = 5, z_2 = \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

B. $L = 2,\ n_{hidden} = 7, z_2 = \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

C. $L = 2,\ n_{hidden} = 5, z_2 = v_2 \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

D. $L = 1,\ n_{hidden} = 7, z_2 = v_2 \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

E. $L = 2,\ n_{hidden} = 5, z_2 = \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

F. $L = 2,\ n_{hidden} = 7, z_2 = v_2 \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

Introduction to Neural Networks

Suppose you want to predict how many minutes per day your cat may ignore you. You consider these input features: 1) number of treats given x_1; 2) time spent petting x_2; 3) presence of a dog at home x_3; 4) presence of a cardboard box at home x_4; and 5) type of the food your cat has x_5. Let y denote the true output value and $\hat{y}$ denote the prediction. Suppose you use a neural network that predicts $\hat{y}$ as follows

$$\hat{y} = \sum_{j=1}^{7} v_j \phi\left(\sum_{i=1}^{5} w_{i,j} x_i + b_j\right) + b$$

where v_j, $w_{i,j}$, b_j, b are trainable parameters and ϕ is an activation function

Let L denote the number of layers, n_{hidden} denote the number of hidden nodes (neurons in hidden layers), and z_2 denote the output (activation) of neuron 2 in hidden layer 1. What are L, n_{hidden}, z_2?

G. $L = 1,\ n_{hidden} = 5, z_2 = \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

H. $L = 2,\ n_{hidden} = 7, z_2 = \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

I. $L = 2,\ n_{hidden} = 5, z_2 = v_2 \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

J. $L = 1,\ n_{hidden} = 7, z_2 = v_2 \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

K. $L = 2,\ n_{hidden} = 5, z_2 = \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

L. $L = 2,\ n_{hidden} = 7, z_2 = v_2 \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

Solution: B

One hidden layer, two layers, 7 hidden nodes, and $z_2 = \phi\left(\sum_{i=1}^{5} w_{i,2} x_i + b_2\right)$

CNNs

A two-dimensional convolutional layer applies filters with spatial size (3, 3), stride 2, and standard padding to an input of size (28,28). What will be the spatial output size?

A. (14,14)
B. (16,16)
C. (28,28)
D. (7,7)
E. (16,16,2)
F. (7,7,2)

CNNs

A two-dimensional convolutional layer applies filters with spatial size (3, 3), stride 2, and standard padding to an input of size (28,28). What will be the spatial output size?

G. (14,14)
H. (16,16)
I. (28,28)
J. (7,7)
K. (16,16,2)
L. (7,7,2)

Solution: A

For standard padding we have, $N^{[2]} = floor\,(\frac{N^{[1]}-1}{2} + 1)$

Then we have $N^{[2]} = floor\,(\frac{28-1}{2} + 1) = 14$

The output spatial size is (14,14)

Deep Architecture Evolution

What is true for “Dropout Regularization”?

A. If a neuron is active and its activation dominates the output of the network, dropout ensures its activation throughout training
B. Dropout removes neurons out of the network at inference time
C. Dropout removes some specific neurons throughout the entire training randomly chosen at initialization
D. Dropout is proposed to improve generalization

Deep Architecture Evolution

What is true for “Dropout Regularization”?

E. If a neuron is active and its activation dominates the output of the network, dropout ensures its activation throughout training
F. Dropout removes neurons out of the network at inference time
G. Dropout removes some specific neurons throughout the entire training randomly chosen at initialization
H. Dropout is proposed to improve generalization

Solution: D

Dropping a neuron out, temporarily remove it from the network at training time
No neuron dominates the output

Distribution Shifts

Suppose you are developing a deep learning model to classify documents as either “advertisement” or “not advertisement” based on various features such as the sender’s information and the presence of certain keywords. Within your training date, you notice that 20% of documents are advertisements. For the test data, you notice that the proportion of ad documents has increased significantly to 50% because of a shift in the collection process of documents. Meanwhile, the distribution of input features given advertisements and not advertisement labels does not change. What is the type of distribution shift you are facing in this example?

A. Covariate Shift
B. Label Shift
C. Class-conditional Shift
D. Full distribution Shift

Distribution Shifts

Suppose you are developing a deep learning model to classify documents as either “advertisement” or “not advertisement” based on various features such as the sender’s information and the presence of certain keywords. Within your training date, you notice that 20% of documents are advertisements. For the test data, you notice that the proportion of ad documents has increased significantly to 50% because of a shift in the collection process of documents. Meanwhile, the distribution of input features given advertisements and not advertisement labels does not change. What is the type of distribution shift you are facing in this example?

E. Covariate Shift
F. Label Shift
G. Class-conditional Shift
H. Full distribution Shift

Solution: B

The proportion of classes has been changed but the distribution of features given advertisements and not advertisement labels has not changed

Performance Estimation

How do you interpret a 95% Confidence Interval for accuracy?

A. On average, 95% of created confidence intervals contain the true accuracy
B. We have run an experiment 100 times, and estimated accuracy inside this interval in 95 of these experiments
C. There is a 95% chance that the true accuracy is within this interval

Performance Estimation

How do you interpret a 95% Confidence Interval for accuracy?

A. On average, 95% of created confidence intervals contain the true accuracy
B. We have run an experiment 100 times, and estimated accuracy inside this interval in 95 of these experiments
C. There is a 95% chance that the true accuracy is within this interval

Solution: A
We don't have to run an experiment 100 times to get a confidence interval. Given a confidence interval, it will either contain the true accuracy or not – there is no probability involved in that.

Vision Transformers

Find the true statement(s) about Swin Transformers

a. Swin Transformers use convolutions for down-sampling

b. Swin Transformers introduce convolutional attention

c. Swin Transformers create hierarchical features

d. Swin Transformers use shifting windows to perform full global attention

Vision Transformers

Find the true statement(s) about Swin Transformers

a. Swin Transformers use convolutions for down-sampling

b. Swin Transformers introduce convolutional attention

c. Swin Transformers create hierarchical features

d. Swin Transformers use shifting windows to perform full global attention

Solution: **C**
Swin Transformers build on Transformers, but merge patches in between transformer blocks. This creates hierarchical features. They do not use convolution. They use shifting windows, but do not perform fully global attention between all patches: it is limited by the windows.
Correct answer is c) Swin Transformers create hierarchical features.

Image Segmentation

What is true about unpooling?

A. Increases resolution
B. Decreases resolution
C. Often used in encoders
D. Increases receptive field
E. Requires a corresponding max-pooling operation

Image Segmentation

What is true about unpooling?

F. Increases resolution
G. Decreases resolution
H. Often used in encoders
I. Increases receptive field
J. Requires a corresponding max-pooling operation

Solution: **A+E**
Unpooling is the reverse operation of a max-pooling, and needs indices from the max-pooling operation to place values back into the correct locations. It is an upsampling technique and therefore increases the resolution. It is often used in the decoder part of a network.

Object detection

Which aspects accurately describe the Detection Transformer (DETR) approach?

A. It relies on predefined anchor boxes.
B. It uses a CNN backbone followed by a Transformer encoder–decoder architecture.
C. It requires Non-Maximum Suppression (NMS) in post-processing to remove duplicate predictions.
D. It uses a fixed number of learned “object queries” in the decoder to predict objects.

Object detection

Which aspects accurately describe the Detection Transformer (DETR) approach?

E. It relies on predefined anchor boxes.
F. It uses a CNN backbone followed by a Transformer encoder–decoder architecture.
G. It requires Non-Maximum Suppression (NMS) in post-processing to remove duplicate predictions.
H. It uses a fixed number of learned “object queries” in the decoder to predict objects.

Solution: B, D
DETR uses neither anchor boxes nor NMS.

Deep Learning Theory Lecture

Let X be a discrete input random variable to some neural network f. Let Y be a discrete random variable representing the target to be learned. Let T = f(X) be the random variable representing the prediction of the network. What do we know about I(T;Y)? Choose two answers.

A. It can be computed by H(Y) – H(Y | T).
B. It can be computed by H(X) – H(T | Y).
C. It measures how good the network compresses the input, the higher it is the more compression we have in the network
D. If it is much lower than I(X;Y), the networks output contains less information about Y than the input

Deep Learning Theory Lecture

Let X be a discrete input random variable to some neural network f. Let Y be a discrete random variable representing the target. Let T = f(X) be the random variable representing the prediction of the network. What do we know about I(T;Y)? Choose two answers.

E. It can be computed by H(Y) – H(Y | T).
F. It can be computed by H(X) – H(T | Y).
G. It measures how good the network compresses the input, the higher it is the more compression we have in the network
H. If it is much lower than I(X;Y), the networks output contains less information about Y than the input

Solution A, D

A is a representation of mutual information discussed in the lecture, D is the definition applied to the setting. B is a wrong formula, and C is wrong as the mutual information between prediction and target variable is not connected to compression.

Backpropagation and Optimization

In a step of gradient descent, if one weight has a very small gradient, what does this mean? Choose one answer.

A The weight is not important for the model

B The loss barely changes when slightly changing the weight

C The weight is already nearly optimal

D The model is underfitting

E The model is overfitting

F The weight should be removed from the model

Backpropagation and Optimization

In a step of gradient descent, if one weight has a very small gradient, what does this mean? Choose one answer.

A The weight is not important for the model

B The loss barely changes when slightly changing the weight

C The weight is already nearly optimal

D The model is underfitting

E The model is overfitting

F The weight should be removed from the model

Solution B

RNN

Which advantage has multilayer RNNs compared to standard one-layer RNNs? Choose one answer.

A. They suffer to a lesser extent from the vanishing gradient problem
B. They suffer to a lesser extent from the exploding gradient problem
C. They can express more complex dependencies between input and output
D. They improve the training stability, allowing faster training.

RNN

Which advantage has multilayer RNNs compared to standard one-layer RNNs? Choose one answer.

E. They suffer to a lesser extent from the vanishing gradient problem
F. They suffer to a lesser extent from the exploding gradient problem
G. They can express more complex dependencies between input and output
H. They improve the training stability, allowing faster training.

Solution C

More layers allow more complex models to fit, but do not improve training performance in general. Also they do not solve exploding/vanishing gradients

Data Augmentation

What can data augmentation not be used for? Choose two answers.

A. Remove outliers from the dataset
B. Gather more training samples without the need to collect them manually.
C. Adjust the distribution of the training data to a distribution also considering the augmented versions of the dataset.
D. Improve the generalization ability of the model.
E. Prevent overfitting
F. Augment each sample with different techniques to artificially create new independent samples from other augment samples.

Data Augmentation

What can data augmentation not be used for? Choose two answers.

G. Remove outliers from the dataset
H. Gather more training samples without the need to collect them manually.
I. Adjust the distribution of the training data to a distribution also considering the augmented versions of the dataset.
J. Improve the generalization ability of the model.
K. Prevent overfitting
L. Augment each sample with different techniques to artificially create new independent samples from other augment samples.

Solution A, F

Targeting outliers is not the goal of data augmentation. We cannot expect independent samples from augmentation techniques.

Tensor shapes

What single layer is missing before the Linear layer?

```
model = nn.Sequential(
        nn.Conv2d(3, 16, 3, padding=1),
        nn.ReLU(),
        nn.MaxPool2d(2),
        nn.Linear(16*16*16, 10)
)
```

A. nn.BatchNorm2d(16)
B. nn.Dropout(0.5)
C. nn.Flatten()
D. nn.Softmax(dim=1)

Tensor shapes

What single layer is missing before the Linear layer?

```
model = nn.Sequential(
        nn.Conv2d(3, 16, 3, padding=1),
        nn.ReLU(),
        nn.MaxPool2d(2),
        nn.Linear(16*16*16, 10)
)
```

A. nn.BatchNorm2d(16)
B. nn.Dropout(0.5)
C. nn.Flatten()
D. nn.Softmax(dim=1)

Solution: C

After Conv2d and MaxPool2d, the tensor is still 4D: [B, C, H, W]. nn.Linear expects a 2D tensor [B, features], so the feature map must be flattened first.

Architectures

What does this code implement?

```
class Block(nn.Module):
        def __init__(self, c):
        super().__init__()
        self.f = nn.Sequential(F
        nn.Conv2d(c, c, 3, padding=1),
        nn.ReLU(),
        nn.Conv2d(c, c, 3, padding=1),
        )

        def forward(self, x):
        return x + self.f(x)
```

A. Transformer encoder
B. Residual/skip connection block
C. GRU cell
D. U-Net decoder

Architectures

What does this code implement?

```
class Block(nn.Module):
        def __init__(self, c):
        super().__init__()
        self.f = nn.Sequential(
        nn.Conv2d(c, c, 3, padding=1),
        nn.ReLU(),
        nn.Conv2d(c, c, 3, padding=1),
        )

        def forward(self, x):
        return x + self.f(x)
```

A. Transformer encoder
B. Residual/skip connection block
C. GRU cell
D. U-Net decoder

Solution: B

The forward pass returns x + F(x). This residual addition is the key idea behind residual or skip-connection blocks, such as those used in ResNet.

Sequence training

You have caption indices [B, T] including <start> and <pad>. Which is the correct input/target split for teacher forcing?

A. inp = captions[:, 1:], tgt = captions[:, :-1]
B. inp = captions[:, :-1], tgt = captions[:, 1:]
C. inp = captions, tgt = captions
D. inp = captions[:, :1], tgt = captions[:, :1]

Sequence training

You have caption indices [B, T] including <start> and <pad>. Which is the correct input/target split for teacher forcing?

A. inp = captions[:, 1:], tgt = captions[:, :-1]
B. inp = captions[:, :-1], tgt = captions[:, 1:]
C. inp = captions, tgt = captions
D. inp = captions[:, :1], tgt = captions[:, :1]

Solution: B

Teacher forcing feeds the current token as input and trains the model to predict the next token. Therefore, inputs are captions without the last token and targets are captions without the first token.