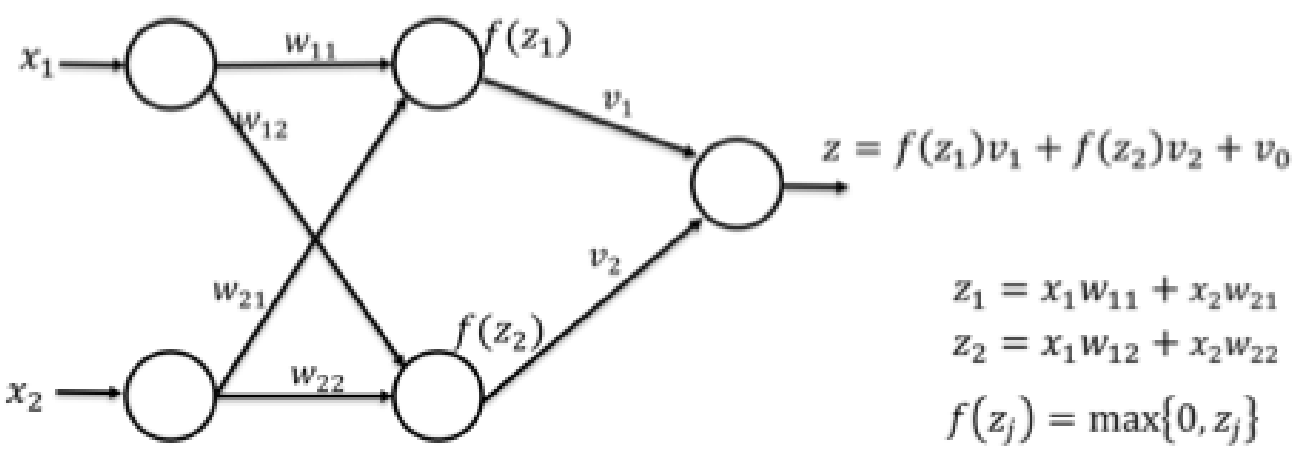
Problem 5

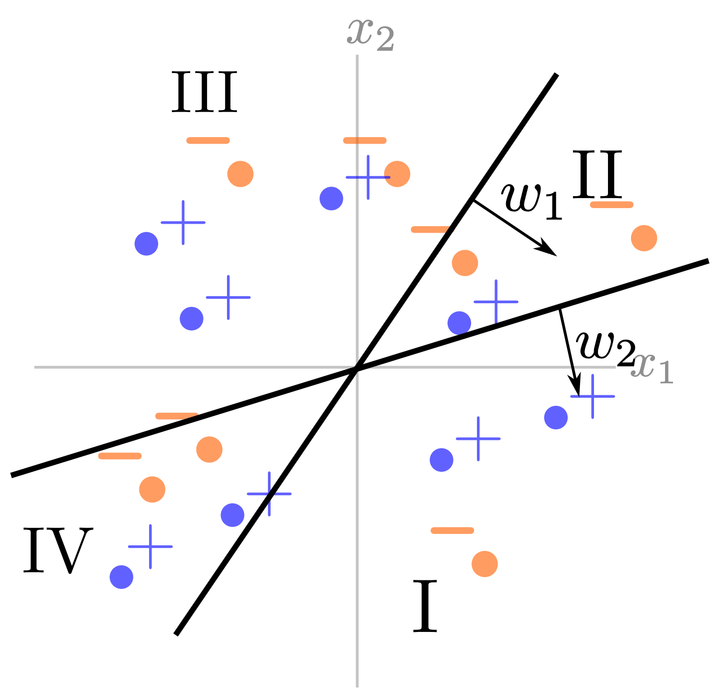
Consider a 2-layer feed-forward neural network that takes in $x \in \mathbb{R}^2$ and has two ReLU hidden units as defined in the figure below. **Note that hidden units have no offset parameters in this problem.**



5. (1)

2/4 points (graded)

The values of the weights in the hidden layer are set such that they result in the z_1 and z_2 "classifiers" as shown in the (x_1, x_2)**-space** in the figure below:



The z_1 "classifier" with the normal $w_1 = [w_{11} \quad w_{21}]^T$ is the line given by $z_1 = x \cdot w_1 = 0$.
Similarly, the z_2 "classifier" with the normal $w_2 = [w_{12} \quad w_{22}]^T$ is the line given by $z_2 = x \cdot w_2 = 0$.
The arrows labeled w_1 and w_2 point in the **positive** directions of the respective normal vectors.
The regions labeled I, II, III, IV are the 4 regions defined by these two lines not including the boundaries.

Choose the region(s) in (x_1, x_2) space which are mapped into each of the following regions in (f_1, f_2)-space, the 2-dimensional space of hidden unit activations $f(z_1)$ and $f(z_2)$. (For example, for the second column below, choose the region(s) in (x_1, x_2) space which are mapped into the f_1-axis in (f_1, f_2)-space.)

(Choose all that apply for each column.)

$\{(f_1, f_2) : f_1 > 0,\ f_2 > 0\}$: f_1-axis: f_2-axis: the origin $(f_1, f_2) = (0, 0)$:

(Choose all that apply.)

☑ I ✔	☐ I	☐ I	☐ I
☐ II	☑ II ✔	☐ II	☐ II
☐ III	☐ III ✔	☐ III ✔	☑ III ✔
☐ IV	☐ IV	☑ IV ✔	☐ IV
☐ None of the above	☐ None of the above	☐ None of the above	☐ None of the above
✔	✖	✖	✔

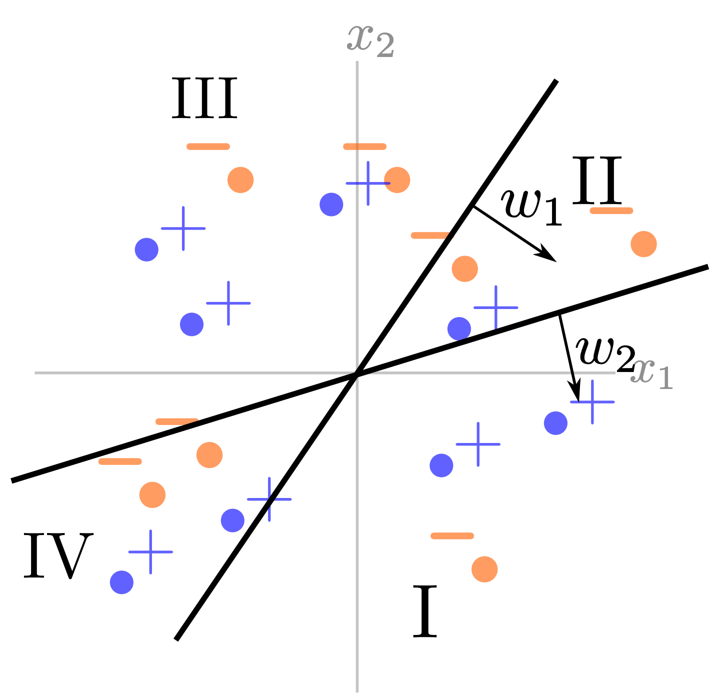
日了，原点也在坐标轴上，服了

Correction Note (July 30 03:00UTC): In an earlier version, the problem statement did not include the emphasis “in the (x_1, x_2)**-space** " in the first sentence.

Correction Note (August 1 13:00UTC): In an earlier version, the caption under the figure did not include “not including the boundaries".

Correction Note (August 4 04:00UTC): Added “(For example, for the second column below, choose the region(s) in (x_1, x_2) space which are mapped into the f_1-axis in (f_1, f_2)-space.)"

Solution:



The regions $\mathrm{I, II, III, IV}$ are defined by (but do not include) the lines $z_1 = x \cdot w_1 = 0$ and $z_2 = x \cdot w_2 = 0$. Hence,

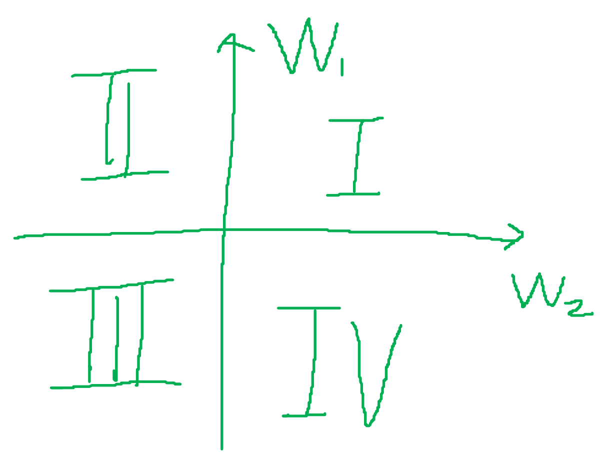
$$z_1 \begin{cases} > 0 & \text{in I, II} \\ < 0 & \text{in I, II} \end{cases}$$

$$z_2 \begin{cases} > 0 & \text{in I, IV} \\ < 0 & \text{in II, III} \end{cases}$$

Applying the reLu function, we get

$$f_1 = f(z_1) \begin{cases} > 0 & \text{in I, II} \\ = 0 & \text{in III, IV} \end{cases}$$

$$f_2 = f(z_2) \begin{cases} > 0 & \text{in I, IV} \\ = 0 & \text{in II, III} \end{cases}.$$



Hence

- The region I in (x_1, x_2)-space maps into the region $\{(f_1, f_2) : f_1 > 0,\ f_2 > 0\}$: in (f_1, f_2)-space ;
- The regions $\mathrm{II,III}$ maps into the region $\{(f_1, f_2) : f_1 > 0,\ f_2 = 0\}$, which is the f_1-axis in (f_1, f_2)-space ;
- The regions $\mathrm{III,IV}$ maps into the region $\{(f_1, f_2) : f_2 > 0,\ f_1 = 0\}$, which is the f_2-axis in (f_1, f_2)-space ;
- The regions III maps to $\{(f_1, f_2) : f_2 = 0,\ f_1 = 0\}$, the origin in (f_1, f_2)-space.

Submit You have used 2 of 3 attempts

Answers are displayed within the problem

5. (2)

2/2 points (graded)

If we keep the hidden layer parameters above fixed but add and train additional hidden layers (applied after this layer) to further transform the data, could the resulting neural network solve this classification problem?

- ( ) yes
- (•) no ✔

Suppose we stick to the 2-layer architecture but add many more ReLU hidden units, all of them without offset parameters. Would it be possible to train such a model to perfectly separate these points?

- (•) yes ✔
- ( ) no

Solution:

- Since points with different labels, namely those in region III, are all mapped to the origin, it is impossible to classify these correctly by only adding more hidden layers.
- The points in regions I,II, IV can be assumed to be distinguished by either the f_1 or f_2 coordinates, so we only need to separate the points in region III, and adding more units, which correspond to more lines through the origin, will work.

Submit You have used 2 of 3 attempts

Answers are displayed within the problem

5. (3)

5/5 points (graded)

Which of the following statements is correct?

1. The gradient calculated in the backpropagation algorithm consists of the partial derivatives of the loss function with respect to each network weight.

   - (•) True ✔
   - ( ) False

2. Initialization of the parameters is often important when training large feed-forward neural networks.

   If weights in a neural network with sigmoid units are initialized all the weights to close to zero values, then during early stochastic gradient descent steps, the network represents a nearly linear function of the inputs.

   - (•) True ✔
   - ( ) False

3. On the other hand, if we randomly set all the weights to very large values, or don't scale them properly with the number of units in the layer below, then the sigmoid units would behave like sign units.

   (Note that a sign unit is a unit with activation function $\mathrm{sign}(x) = 1$ if $x > 0$ and $\mathrm{sign}(x) = -1$ if $x < 0$. For the purpose of this question, it does not matter what $\mathrm{sign}(0)$ is.)

   - (•) True ✔
   - ( ) False ✔

   Grading Note: Since the question did not specify whether "behave like sign units" allows for shifting or rescaling of the sign function, both "True" and "False" are accepted as correct.

4. If we use only sign units in a feedforward neural network, then the stochastic gradient descent update will

   - (•) almost never change any of the weights ✔
   - ( ) change the weights by large amounts at random

5. Stochastic gradient descent differs from (true) gradient descent by updating only one network weight during each gradient descent step.

   - ( ) True
   - (•) False ✔

Correction note (July 31 16:00UTC):. In the earlier version, the sign unit definition was not included.

Solution:

- True, by definition of the backpropagation algorithm.
- True, because those activation functions are linear-like near zero.
- True, because far from zero the sigmoid function looks like a scaled and shifted version of the sign function.
- True, because of zero gradient.
- False. Stochastic gradient descent differs by considering the gradient with respect to just one training example on each step.

Submit You have used 2 of 3 attempts

Answers are displayed within the problem

5. (4)

3/3 points (graded)

There are many good reasons to use convolutional layers in CNNs as opposed to replacing them with fully connected layers. Please check T or F for each statement.

Since we apply the same convolutional filter throughout the image, we can learn to recognize the same feature wherever it appears.

- (•) True ✔
- ( ) False

A fully connected layer for a reasonably sized image would simply have too many parameters

- [x] True ✔
- [ ] False

A fully connected layer can learn to recognize features anywhere in the image even if the features appeared preferentially in one location during training

- [ ] True
- [x] False ✔

Solution:

- True by definition
- True. For a fully connected layer of size n and a picture of size $w \times h$, the size of the layer would be whn, several times larger than the image itself.
- False.Only a convolutional layer can do this. A fully connected layer's weights depend on the location, so they would only be able to recongize features at locations that appeared preferentially in the training data.

Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Error and Bug Reports/Technical Issues

Show Discussion

Topic: Midterm Exam (1 week):Midterm Exam 1 / Problem 5