Quiz chapter 2.5-2.6

Due No due date **Points** 0 **Questions** 8 **Time Limit** None
Allowed Attempts Unlimited

Instructions

These practice quizzes are meant to trigger thoughts for the "open hour" help sessions once a week.

There are "correct" answers specified, but sometimes, answers are not straight forward. It can be discussed during help sessions.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	6 minutes	0 out of 0

Submitted Nov 12 at 1:52pm

Question 1 — 0 / 0 pts

Softmax

What is true about the "softmax" output activation function?

You can select any number of options.

- [x] It ensures normalization of outputs: the sum of outputs is 1.
- [ ] It ensures all outputs are between 0 and 1.
- [x] It follows the standard node calculations, in the sense that the node output depends only on the weighted sum of inputs to the same node.

  The softmax activation is different, since the output of node i (y_i) depends on the argument to node i (a_i) but also on all other arguments to the output layer (all $a_{i'}$).

- [x] When combined with Categorical Cross-entropy error as loss, it gives the same Gradient Descent expressions as our standard choices for regression and binary classification.
- [ ] It cannot be used for binary classification, there must be more than 2 classes.

Wrong answer

Question 2 — 0 / 0 pts

XOR data

We have a dataset with 2-diemensional input and a single target representing binary classification.

All patterns with $x_1 \cdot x_2 > 0$ belong to class 1. All other belong to class 0. (The data represents a kind of XOR problem: If one but only one input is positive, we have class 0)

Can we represent this data using an MLP with a single hidden layer?

- ( ) Yes
- (•) No

  Yes, the Universal Approximation Theorem holds.

Correct answer

Question 3 — 0 / 0 pts

XOR function

Instead of data, just consider the binary function of 2 variables

d=1 for $x_1 \cdot x_2 > 0$, and d=0 otherwise.

Can we represent this function using an MLP with a single hidden layer?

- ( ) Yes
- (•) No

  The definition range is for all real x_1 and x_2, so it is not bounded. Furthermore, the target function is discontinuous for any x=0. Because of this, the Universal Approximation Theorem does not apply. It could of course be possible to solve the task anyway, but it turns out it isn't.

Wrong answer

Question 4 — 0 / 0 pts

Back-propagation and plateaus

A potential problem in back-propagation is to get stuck in a "plateau", where the derivative of the loss E w.r.t. an early weight (e.g. input-to-first hidden) is near zero without being close to a minimum. What kind of hidden layer activation functions $\varphi(a)$ do you think are good to avoid vanishing gradients?

- [ ] tanh(a)
- [x] rectifier: max{0,a}
- [ ] logistic: 1/(1+exp(-a))
- [x] threshold-type: sgn(a)

  We let Canvas assign rectifier as the only "good" one, which is misleading. It is meant to be "best", but in many cases, tanh(a) is good enough, and logistic is just a linear modification of tanh(a). The threshold function is not differentiable at all: all gradients vanish, except for a=0 when it is undefined.

Correct answer

Question 5 — 0 / 0 pts

SGD terminology

Select explanations to the terms related to Stochastic Gradient Descent (SGD).

There is one explanation too many.

Epoch	A set of iterations wl
Iteration	One update of the w
Minibatch	A subset of patterns
Batch	The entire dataset
Online update	When a single patter
Pattern	One "data point" wit

Other Incorrect Match Options:

- A new random split of the entire dataset into sub-groups for updates.

Just to make life hard for new students, most implementations of networks avoid shorten the correct term "mini-batch" to just "batch". The entire dataset, that is also called the "batch" in theoretical texts, is never given that name in such implementations.

Correct answer

Question 6 — 0 / 0 pts

SGD vs GD

Which of the following statements fits the stochastic gradient descent method (as compared to the ordinary gradient descent method)?

- [ ] SGD makes training slower
- [x] SGD is better at avoiding local minima
- [ ] SGD gives a more accurate calculation of the gradient in each iteration
- [ ] SGD automatically adjusts the learning rate
- [x] SGD can be problematic for classifications problems with very unbalanced classes

Some comments:

The random selection in a mini-batch can help "shake out" of a local minimum.

Neither SGD nor normal GD adjusts the learning rate automatically.

If very many mini-batches contain a single class, training can face problems. This can be handled for example by making multiple copies of the patterns in the minority class (upsampling).

Correct answer

Question 7 — 0 / 0 pts

Momentum

One gradient descent improvement is called "momentum". What is true about that?

- [x] You add a part of the previous weight update to the current one
- [ ] You dynamically increase the learning rate as you train
- [ ] You increase the minibatch-size as you train

Correct answer

Question 8 — 0 / 0 pts

Bold driver

The "bold driver" method compares losses to dynamically change the learning rate.

If the error increases after a weight update, E(t+1)>E(t), the learning rate is reduced.

If the error reduces, E(t+1)<E(t), the learning rate is increased.

Is "bold driver" suitable to combine with Stochastic Gradient Descent, where a random sub-sample of patterns are used in each weight update?

- ( ) Yes
- (•) No

  In SGD, the losses E(t) and E(t+1) are calculated based on different patterns. The inequalities E(t+1) [> or <] E(t) cannot be interpreted as being close or far from the optimum, and should not be guiding changes in learning rate.