Calvin AI/ML Handout | *Feb 26, 2025*

Name: ______________________________

1. Suppose we're trying to minimize a function $f(w) = w^2 + 2w + 1$ using gradient descent:

```
def f(w):
  return w**2 + 2*w + 1

def grad_f(w):
  # gradient of f with respect to w
  return ______________________
```

```
f(3) = ______________________
grad_f(3) = ______________________
# fill in the blank to minimize f(w)
w = random()
for i in range(100):
  w = ______________________
```

2. Let's use an MLP with one hidden layer to **classify** handwritten digits as 0, 1, ..., 9. We'll give the model a 28x28-pixel image of a digit, flattened into a 784-dimensional vector. Fill in the blanks below with reasonable values.

```
W1 = random_array(size=(______________________))
b1 = random_array(size=(______________________))
W2 = random_array(size=(______________________))
b2 = random_array(size=(______________________))

for x_batch, y_batch in training_data:
  # forward pass, starting with linear layer
  linear_out_1 = ______________________________
  activations_1 = _____________________________
  logits = ______________________________
  probs = ______________________________
  loss = cross_entropy_loss(probs, y_batch)

  # backward pass
  loss.backward() # grads now stored in .grad
  for param in [W1, b1, W2, b2]:
    param += _________________________
```

```
example_x = x_batch[0]
example_y = y_batch[0] # see y_batch.shape
example_x = [________________________________]
example_y = [________________________________]

# Fill in the shapes below:
x_batch.shape = (N, 784); y_batch.shape = (N,)
linear_out_1.shape = _________________________
activations_1.shape = _______________________
logits.shape = _________________________
probs.shape = ______________________________
loss.shape = _________

W1.grad.shape = _________________________
b1.grad.shape = _________________________
W2.grad.shape = _________________________
b2.grad.shape = _________________________
```

Calvin AI/ML Handout | *Feb 26, 2025*

Name: ______________________________

1. Suppose we're trying to minimize a function $f(w) = w^2 + 2w + 1$ using gradient descent:

```
def f(w):
  return w**2 + 2*w + 1

def grad_f(w):
  # gradient of f with respect to w
  return ______________________
```

```
f(3) = ______________________
grad_f(3) = ______________________
# fill in the blank to minimize f(w)
w = random()
for i in range(100):
  w = ______________________
```

2. Let's use an MLP with one hidden layer to **classify** handwritten digits as 0, 1, ..., 9. We'll give the model a 28x28-pixel image of a digit, flattened into a 784-dimensional vector. Fill in the blanks below with reasonable values.

```
W1 = random_array(size=(______________________))
b1 = random_array(size=(______________________))
W2 = random_array(size=(______________________))
b2 = random_array(size=(______________________))

for x_batch, y_batch in training_data:
  # forward pass, starting with linear layer
  linear_out_1 = ______________________________
  activations_1 = _____________________________
  logits = ______________________________
  probs = ______________________________
  loss = cross_entropy_loss(probs, y_batch)

  # backward pass
  loss.backward() # grads now stored in .grad
  for param in [W1, b1, W2, b2]:
    param += _________________________
```

```
example_x = x_batch[0]
example_y = y_batch[0] # see y_batch.shape
example_x = [________________________________]
example_y = [________________________________]

# Fill in the shapes below:
x_batch.shape = (N, 784); y_batch.shape = (N,)
linear_out_1.shape = _________________________
activations_1.shape = _______________________
logits.shape = _________________________
probs.shape = ______________________________
loss.shape = _________

W1.grad.shape = _________________________
b1.grad.shape = _________________________
W2.grad.shape = _________________________
b2.grad.shape = _________________________
```

Before you leave, pick a couple of these questions to react to:

1. What was the most important concept from today for you?
2. What was the muddiest concept today?
3. How does what we did today connect with what you've learned before?
4. What would you like to review or clarify next time we meet?
5. What are you curious, hopeful, or excited about?

Before you leave, pick a couple of these questions to react to:

1. What was the most important concept from today for you?
2. What was the muddiest concept today?
3. How does what we did today connect with what you've learned before?
4. What would you like to review or clarify next time we meet?
5. What are you curious, hopeful, or excited about?