CS342 - Section 3

- Homework 2
  - Due 9/13, can still turn it in before 9/15!
- Homework 3
  - Will be posted today
  - Classification on SuperTux data you collected!
- Quiz 2
  - Solution posted
  - Grade will be posted by next Mon

Agenda

- Quiz 2 Review
- Regression vs Classification
- Tips for training DNN
- Homework 3

Quiz 2

- Forward propagation
  - Compute model output & loss from input
- Backward propagation
  - Compute gradient wrt hidden values $dl(\hat{y})/dz_i$
  - Compute gradient wrt weights $dl(\hat{y})/dW_i$ using $dl(\hat{y})/dz_i$
- Why calculating gradient backward
  - Smaller matrix multiplication
  - Reuse computation

Quiz 2

- Forward Propagation
  - Given x, find $\hat{y}$ and $l(y, \hat{y})$
  - E.g.1(a)
    i. $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $y = 2$
    ii. $z_1 = \begin{pmatrix} 1 & -2 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
    iii. $z_2 = \text{relu}(z_1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
    iv. $\hat{y} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}^T z_2 = 1$
    v. $l(y, \hat{y}) = \frac{1}{2}(1-2)^2 = \frac{1}{2}$
- Questions

Quiz 2

- Review: Jacobian Matrix
  - Matrix of derivatives for vector-valued function
  - $y \in R^m$, $x \in R^n$, $y=f(x)$
  - $dy/dx = \begin{pmatrix} dy_1/dx_1 & \dots & dy_1/dx_n \\ \dots & & \dots \\ dy_m/dx_1 & \dots & dy_m/dx_n \end{pmatrix}$
  - $y = Wx$, $dy/dx = ?$
  - $y = \text{relu}(x) = ?$

Quiz 2

- Review: Jacobian Matrix
  - $y \in R^m$, $x \in R^n$
  - $dy/dx = \begin{pmatrix} dy_1/dx_1 & \dots & dy_1/dx_n \\ \dots & & \dots \\ dy_m/dx_1 & \dots & dy_m/dx_n \end{pmatrix}$
  - $y = Wx$, $dy/dx = W$
  - $y = \text{relu}(x) = \text{diag}([1\ 0\ 1\ \dots\ 0\ 1])$

Quiz 2

- Backward Propagation
  - Given x, find dl/dz_i
  - E.g.2(a)
    i. $l = \frac{1}{2}(\hat{y}-y)^2 \Rightarrow dl/d\hat{y} = \hat{y} - y$
    ii. $\hat{y} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T z_2 \Rightarrow d\hat{y}/dz_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T$
    iii. $z_2 = \text{relu}(z_1) \Rightarrow dz_2/dz_1 = \text{diag}([1\ 0])$
    iv. $z_1 = \begin{pmatrix} 1 & -2 \\ 0 & 2 \end{pmatrix} x \Rightarrow dz_2/dx = \begin{pmatrix} 1 & -2 \\ 0 & 2 \end{pmatrix}$

Quiz 2

- Backward Propagation
  - Given x, find dl/dz_i
  - E.g.2(a)
    i. $dl/dx = dl/\hat{y} * d\hat{y}/dz_2 * dz_2/dz_1 * dz_1/dx$
    ii. $dl/dx = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$

Quiz 2

- Backward Propagation
  - Given x, find dl/dz_i
  - E.g.2(a)
    i. Verify in PyTorch

```
import torch

class Eg(torch.nn.Module):
    def __init__(self):
        super(Eg, self).__init__()
        self.fc1 = torch.nn.Linear(2,2)
        self.relu = torch.nn.ReLU()
        self.fc2 = torch.nn.Linear(2,1)

        self.fc1.weight = torch.nn.Parameter(torch.tensor([[1.,-2.],[0.,2.]]))
        self.fc1.bias = torch.nn.Parameter(torch.tensor([0.,-1.]))
        self.fc2.weight = torch.nn.Parameter(torch.tensor([[1.,-1.]]))
        self.fc2.bias = torch.nn.Parameter(torch.tensor([0.]))

    def forward(self, x):
        self.z1 = self.fc1(x)
        self.z2 = self.relu(self.z1)
        out = self.fc2(self.z2)
        return out

x =torch.tensor( [1.,0.] , requires_grad=True)
y = 2
m = Eg()
out = m(x)
loss = 1/2*(y-out)**2
loss.backward()
print (x.grad)
```

Quiz 2

- Why backpropagation?
  - Note
    i. Computing (dot) product of two vectors of size l requires l multiplications and l-1 additions
    ii. Computing product of two matrices of size kxl and lxm requires km(2l-1) operations
  - E.g 3
    i. 1x2+3x2+3x2=14 total operations
  - E.g 4
    i. 3x4+3x2+1x2=20 total operations
    ii. Cannot reuse info to compute gradient wrt W, need to compute all: 20 operations
- Question for Quiz 2

Regression vs Classification

- Recap from lectures
  a. Regression: predict continuous value
  b. Classification: predict class label
- In homework 1, people used different loss functions
  a. Number label + L2 loss (NEVER DO THIS)
  b. One hot label + L2 loss
  c. One hot label + cross entropy loss

Regression vs Classification

- Number label + L2 loss
  - Bias in gradient; If ground truth is 0, “more wrong” to predict 2 than 1?. Using this loss assumes order in classes.
- One hot label + Cross entropy loss
  - Equivalent to maximizing likelihood
- One hot label + L2 loss
  i. Actually has a name called “Brier score”: $\sum(p_k - o_k)^2$, $o_k \in \{0,1\}$
  ii. More gentle than cross entropy loss. E.g. If predicted [0,1] but actually [1,0], cross entropy loss is infinite whereas brier score is finite
  iii. People have found that this loss converge slower

Tips for training NN (for now)

- Display your loss every T gradient step
  - Make sure loss is “sane”, e.g. decreasing and not blowing up
  - Determine when to stop training
- Visualize model output
  - Image segmentation
  - Decision boundary
- Make sure data is correct

Homework 3

- Starter code clarification