

Introduction to Machine Learning

Day 13: Regularization for Neural Networks - Early Stopping and Dropout

Joanna Bieri DATA301

Get out your handwritten notes!

Mini Review - The Setup

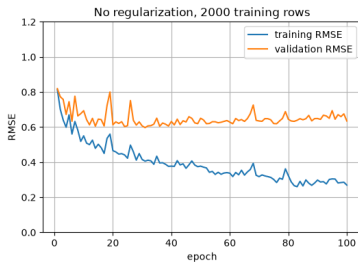
REVIEW

California housing again, but only the first **2000** training rows. The validation set is the same 5160 rows.

A network with 3 hidden layers of 100 neurons has about 21,000 numbers to learn, so it has plenty of room to memorize 2000 rows.

Mini Review - Overfitting

REVIEW



Best validation RMSE 0.5972 at epoch 31. After 100 epochs: training 0.2698, validation 0.6353.

Mini Review - Early Stopping

REVIEW

- Check the validation RMSE after every epoch.
- Keep a copy of the weights from the best epoch.
- Stop when there has been no new best for **patience** epochs (we used 20).

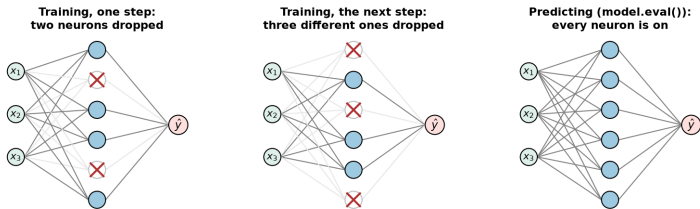
```
best_weights = copy.deepcopy(model.state_dict())
```

```
stopped after epoch 51
```

```
best epoch: 31    validation RMSE: 0.5972
```

Mini Review - Dropout

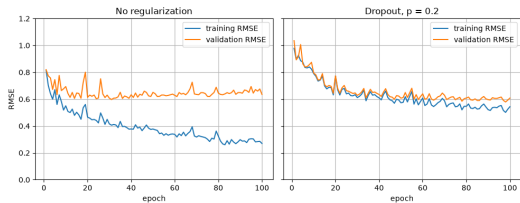
REVIEW



At every training step, each neuron's output is set to 0 with probability p , the **dropout rate**.

Mini Review - With and Without Dropout

REVIEW



After 100 epochs, no regularization: training 0.2698, validation 0.6353.
Dropout 0.2: training 0.5435, validation 0.6079.

Mini Review - Weight Decay

REVIEW

Ridge regression from Day 3, for a neural network. It is one argument to the optimizer:

```
torch.optim.Adam(model.parameters(), lr=0.001,  
                  weight_decay=0.0001)
```

The Whole Process - 1. The model

CODE

```
model = nn.Sequential(  
    nn.Linear(8, 100), nn.ReLU(),  
    nn.Dropout(p=0.2),          # dropout after the activation  
    nn.Linear(100, 100), nn.ReLU(),  
    nn.Dropout(p=0.2),  
    nn.Linear(100, 100), nn.ReLU(),  
    nn.Dropout(p=0.2),  
    nn.Linear(100, 1))          # no dropout after the output
```

The Whole Process - 2. The starting weights

CODE

```
for layer in model:  
    # only the Linear layers have weights  
    if isinstance(layer, nn.Linear):  
        nn.init.kaiming_normal_(layer.weight,  
                                nonlinearity="relu")  
        nn.init.zeros_(layer.bias)
```

The Whole Process - 3. Batches, loss and optimizer

CODE

```
train_loader = DataLoader(train_data, batch_size=32,  
                           shuffle=True)  
  
loss_fn = nn.MSELoss()  
  
optimizer = torch.optim.Adam(model.parameters(),  
                              lr=0.001)
```

The Whole Process - 4. Set up early stopping

CODE

```
patience = 20
best_rmse = float("inf")      # bigger than any number
best_weights = None
epochs_without_improving = 0
```

The Whole Process - 5. Train one epoch

CODE

```
for epoch in range(200):          # the most epochs we will train
    model.train()                  # training mode: dropout is ON
    for X_batch, y_batch in train_loader:
        optimizer.zero_grad()
        y_pred = model(X_batch)
        loss = loss_fn(y_pred, y_batch)
        loss.backward()
        optimizer.step()
```

The Whole Process - 6. Check the validation RMSE

CODE

```
model.eval()                                # prediction mode: dropout is OFF
with torch.no_grad():
    rmse = torch.sqrt(loss_fn(model(X_valid_t),
                                   y_valid_t)).item()
```

The Whole Process - 7. Keep the best, or count

CODE

```
if rmse < best_rmse:          # a new best, save a copy
    best_rmse = rmse
    best_epoch = epoch + 1
    best_weights = copy.deepcopy(
                        model.state_dict())
    epochs_without_improving = 0
else:
    epochs_without_improving = \
        epochs_without_improving + 1
```

The Whole Process - 8. Stop, and put the best weights back

CODE

```
if epochs_without_improving == patience:  
    print("stopped after epoch", epoch + 1)  
    break  
  
model.load_state_dict(best_weights)
```

Overfitting

ASK

How would you know if a neural network
is overfitting? Say it in words.

Overfitting

ANSWER

How would you know if a neural network is overfitting?

- Plot the learning curves: the training RMSE keeps going down, but the validation RMSE stops going down or goes back up.
- There is a big gap between the training RMSE and the validation RMSE.
- Warning sign before you even train: lots of parameters and not much data.

Early Stopping

ASK

Why can we not decide when to stop
by watching the training RMSE?

Early Stopping

ANSWER

Why can we not decide when to stop by watching the training RMSE?

- The training RMSE keeps going down almost forever, even while the model gets worse on new rows.
- Only the validation RMSE shows when the model stops getting better on new data.

Early Stopping

ASK

Early stopping waits for a new best for "patience" epochs before it stops.

Why wait at all?

What if the patience was 1?

Early Stopping

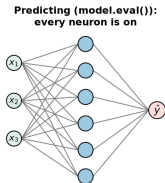
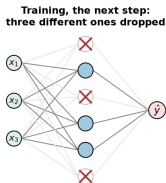
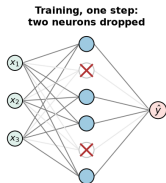
ANSWER

Why wait at all? What if the patience was 1?

- The validation RMSE bounces up and down from epoch to epoch.
- One bad epoch does not mean the model is done improving.
- With a patience of 1 we would stop at the first bad epoch, long before the real best.

Dropout

ASK



In your own words, what does dropout do while we train? Why does that help on new data?

Dropout

ANSWER

What does dropout do while we train? Why does that help on new data?

- At every step it turns off a random set of neurons, a different set each time.
- The network can not count on one special neuron to memorize a training row.
- It has to spread what it learns out over many neurons, and those patterns work on new rows too.

Dropout

ASK

Why is dropout turned off
when we predict?

Dropout

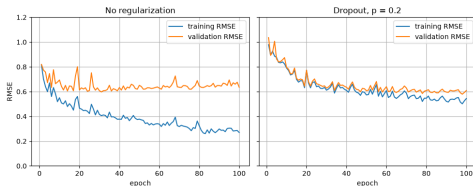
ANSWER

Why is dropout turned off when we predict?

- When we predict we want the whole trained network.
- Random dropout would make the predictions random, a different answer every time.
- `model.eval()` turns it off, `model.train()` turns it back on.

With and Without Dropout

ASK



Same network, same 100 epochs. Dropout made the training RMSE worse, 0.2698 went to 0.5435. Why is that a good thing here?

With and Without Dropout

ANSWER

Dropout made the training RMSE worse. Why is that a good thing here?

- A small training RMSE here meant the network was memorizing.
- Dropout makes memorizing hard, so the training RMSE goes up.
- The validation RMSE got better, 0.6353 to 0.6079, and the gap went from about 0.37 to about 0.06.

Questions?

Questions?