

NLP Assignment2 Task2 Reportf

Configuration of My CNN Model

Structure

First we use the `embedding` to embed the word vectors for each word.
Then we need to create kernels with different sizes to implement the convolution layers.
After that, we apply `ReLU` on the output and choose the maximum.
At last, we concatenate them all and put them into a linear layer.

Code

```
1 class TextCNN(nn.Module):
2     def __init__(self, vocab_size, embedding_dim, kernel_sizes, num_channels):
3         super(TextCNN, self).__init__()
4         self.word_embeddings = nn.Embedding(vocab_size+1, embedding_dim, padding_idx=w2i['<PAD>'])
5         self.dropout = nn.Dropout(0.5)
6         self.decoder = nn.Linear(sum(num_channels), 4)
7         self.pool = GlobalMaxPool1d()
8         self.convs = nn.ModuleList()
9         for c, k in zip(num_channels, kernel_sizes):
10             self.convs.append(nn.Conv1d(in_channels = embedding_dim,
11                                         out_channels = c,
12                                         kernel_size = k))
13
14     def forward(self, sentence):
15         embeds = self.word_embeddings(sentence)
16         embeds = embeds.permute(0, 2, 1)
17         encoding = torch.cat([self.pool(F.relu(conv(embeds))).squeeze(-1) for conv in self.convs], dim=1)
18         outputs = self.decoder(self.dropout(encoding))
19         return outputs
```

Parameters

```
learning rate = 0.001
num_epochs = 100
embedding_dim = 200
kernel_size = [3, 4, 5]
num_channels = [100, 100, 100]
batch_size = 32
loss = CrossEntropyLoss()
```

Classification Accuracy on Test Set

I ran 5 times and counted them, the results are shown below.

Times	Accuracy
1	0.813
2	0.804
3	0.789
4	0.816
5	0.827

the mean and variance are

mean	variance
0.810	1.62e-4