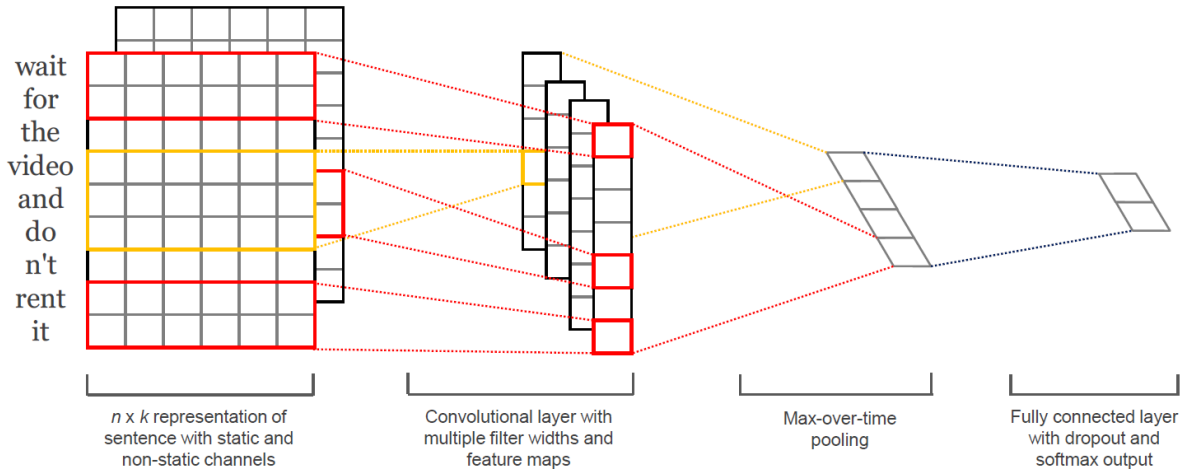
1d Convolution Network for time series

때때로 1d 컨볼루션 네트워크는 RNN보다 더 잘 작동하고 연산 비용도 더 저렴하다. (수브라마니안비슈누, 2019)

(KimYoon, 2014)



이를 보면 시계열 자료를 1d 커널이 이동하면서 패턴을 학습함을 알 수 있다. 그런데 이 경우 단점은 매우 시계열이 분명한 데이터가 아니라면 불필요하게 자료간 연관성을 학습할 수 있다는 점이다.

아래는 1d 컨볼루션 네트워크에 대한 의사코드이다.

```
class 1dCNN(nn.Module):
    def __init__(self, 데이터, hidden_size, n_cat, bs=1, kernel_size=시계열 길이(hyperparameter), max_len=200):
        super().__init__()
        self.hidden_size = hidden_size
        self.bs = bs
        self.e = nn.Embedding(데이터,  hidden_size)
        self.cnn = nn.Conv1d(max_len, hidden_size, kernel_size)
        self.avg = nn.AdaptiveAvgPool1d(100)
        self.fc1 = nn.Linear(1000, n_cat)
        self.fc2 = nn.Linear(n_cat, 1)

        def forward(self, inp):
            bs = inp.size()[0]
            if bs != self.bs:
                self.bs = bs
            e_out = self.e(inp)
            cnn_o = self.cnn(e_out)
            cnn_avg = self.avg(cnn_o)
            fc1 = F.dropout(self.fc1(cnn_avg), p=0.5)
            fc2 = self.fc2(fc1)
        return fc2
```

생각보다 단순한 구조로 커널을 이용해서 시계열 자료들을 적당한 길이로 컨볼루션하고, 이를 FC 레이어들을 통해서 값을 예측하는 방식이다. 그러나, RNN에서의 결과가 좋지 못했듯이 1dCNN의 경우에도 결과가 좋지 못했다. Training loss가 0.22 validation loss가 0.44에서 0.5 사이를 왔다갔다 거리면서 converge하지 못하는 것이 관찰되었다. 오히려 기본 NN으로 하는 것이 더 좋은 결과를

보인 것으로 보아, 팀 내에서는 시계열 경향성보다는 일반적인 분류 기능을 통한 예측이 보다 더 좋은 결과를 보인다고 결론을 내렸다.

참고 문헌

KimYoon. (2014). Convolutional Neural Networks for Sentence Classification. "EMNLP", (페이지: 1747).

수브라마니안비슈누. (2019). "PyTorch로 시작하는 딥러닝." 에이콘.