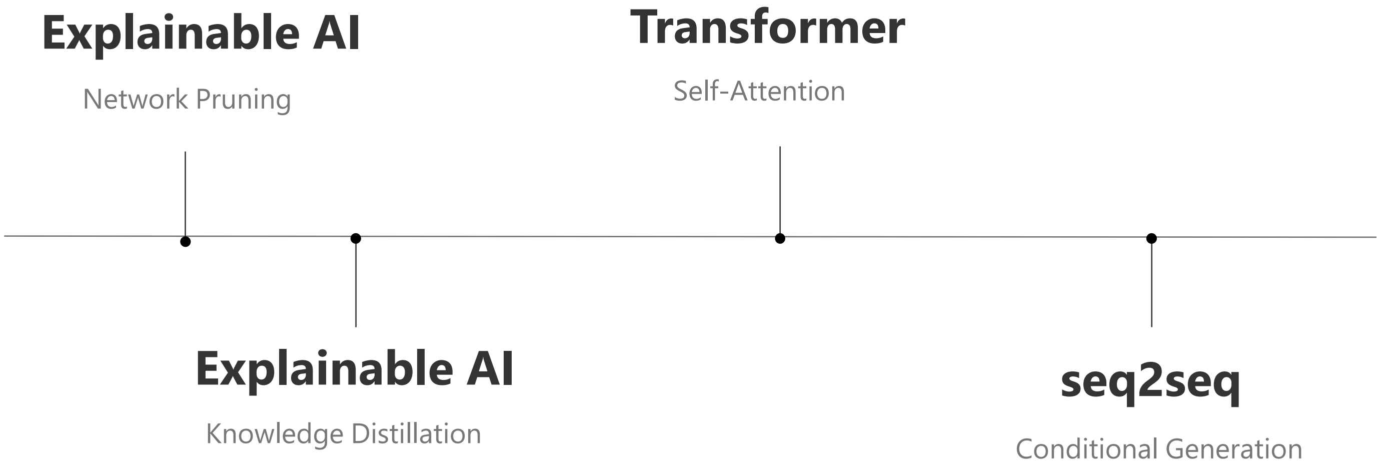
每周总结



Explainable AI

Network Pruning

Explainable AI

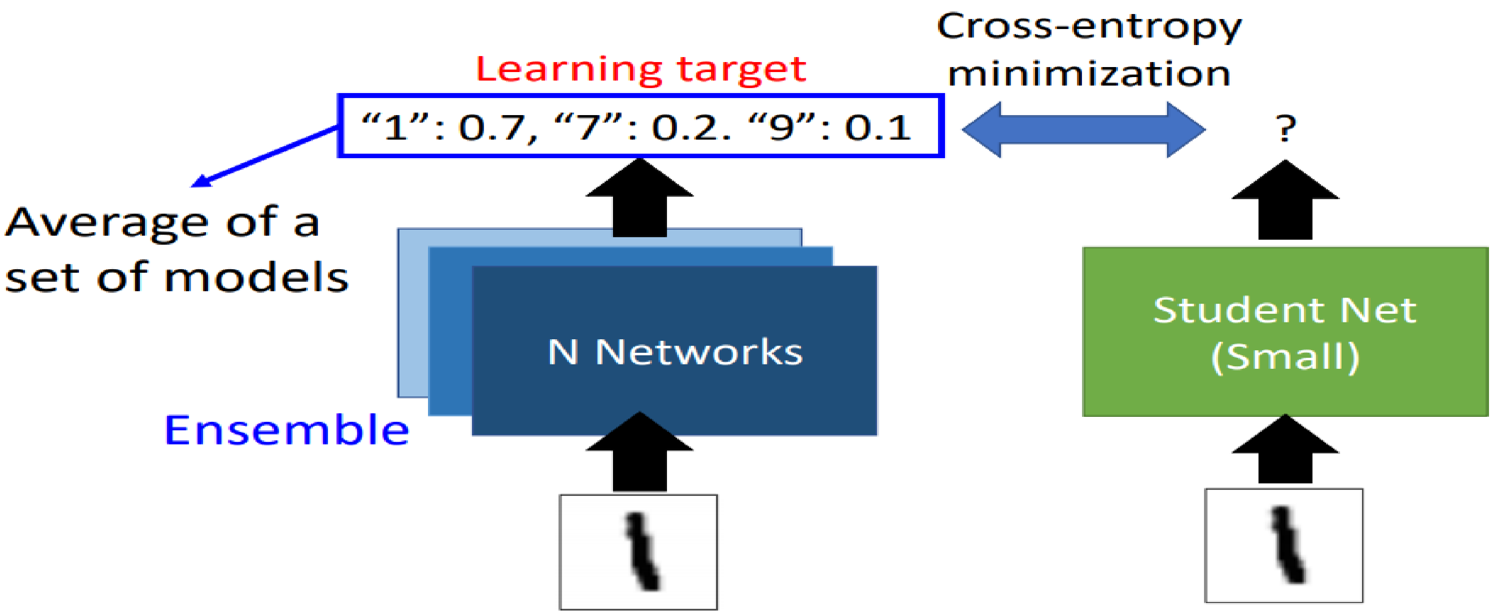
Knowledge Distillation

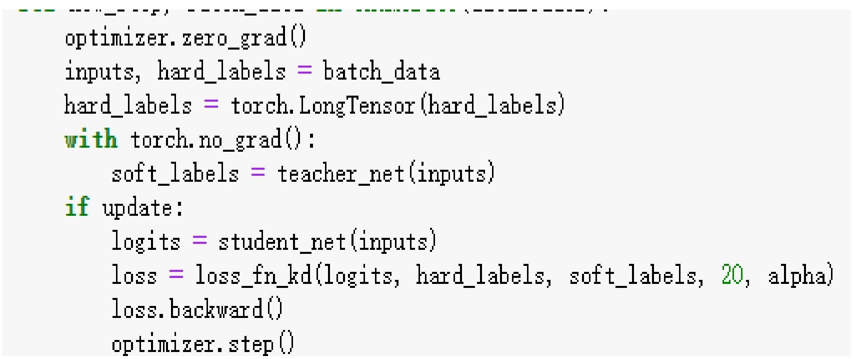
Transformer

Self-Attention

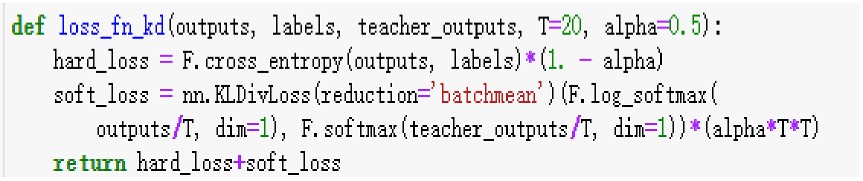
seq2seq

Conditional Generation

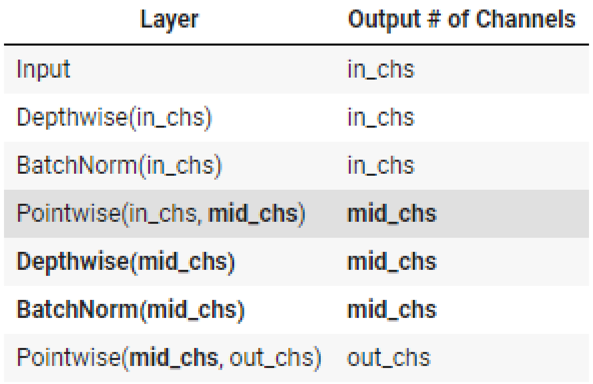




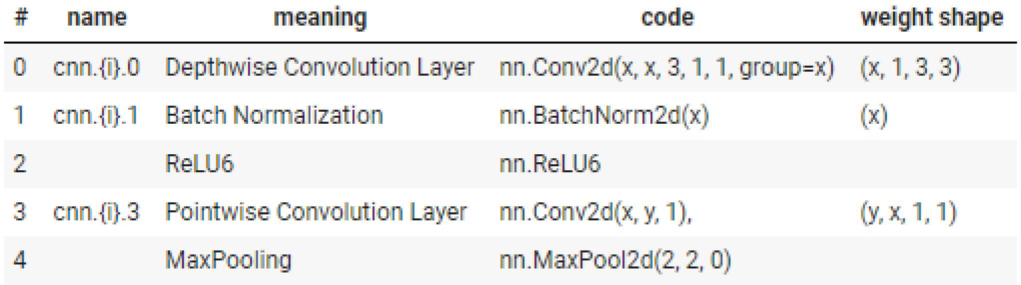
```
    optimizer.zero_grad()
    inputs, hard_labels = batch_data
    hard_labels = torch.LongTensor(hard_labels)
    with torch.no_grad():
        soft_labels = teacher_net(inputs)
    if update:
        logits = student_net(inputs)
        loss = loss_fn_kd(logits, hard_labels, soft_labels, 20, alpha)
        loss.backward()
        optimizer.step()
```



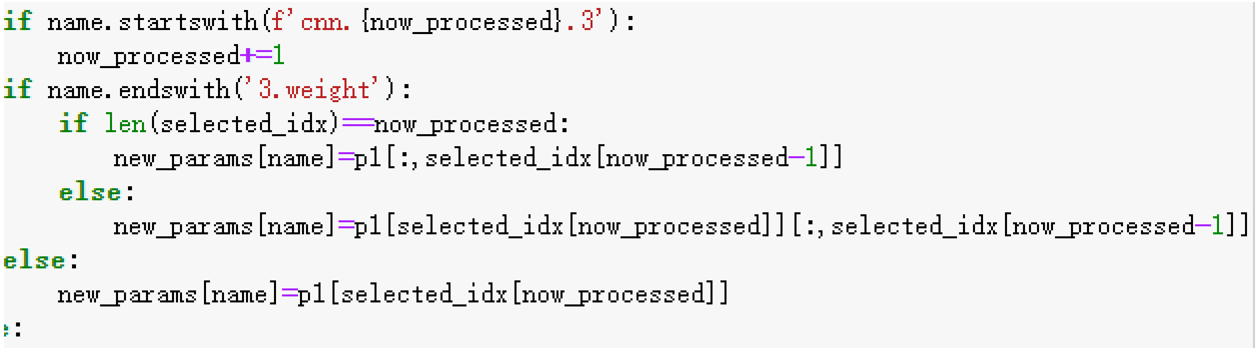
```
def loss_fn_kd(outputs, labels, teacher_outputs, T=20, alpha=0.5):
    hard_loss = F.cross_entropy(outputs, labels)*(1. - alpha)
    soft_loss = nn.KLDivLoss(reduction='batchmean')(F.log_softmax(
        outputs/T, dim=1), F.softmax(teacher_outputs/T, dim=1))*(alpha*T*T)
    return hard_loss+soft_loss
```



Layer	Output # of Channels
Input	in_chs
Depthwise(in_chs)	in_chs
BatchNorm(in_chs)	in_chs
Pointwise(in_chs, **mid_chs**)	**mid_chs**
Depthwise(mid_chs)	**mid_chs**
BatchNorm(mid_chs)	**mid_chs**
Pointwise(**mid_chs**, out_chs)	out_chs

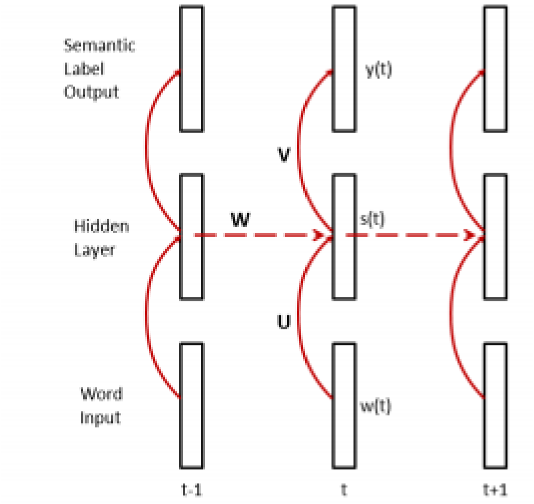


#	name	meaning	code	weight shape
0	cnn.{i}.0	Depthwise Convolution Layer	nn.Conv2d(x, x, 3, 1, 1, group=x)	(x, 1, 3, 3)
1	cnn.{i}.1	Batch Normalization	nn.BatchNorm2d(x)	(x)
2		ReLU6	nn.ReLU6	
3	cnn.{i}.3	Pointwise Convolution Layer	nn.Conv2d(x, y, 1),	(y, x, 1, 1)
4		MaxPooling	nn.MaxPool2d(2, 2, 0)	



```
if name.startswith(f'cnn.{now_processed}.3'):
    now_processed+=1
if name.endswith('3.weight'):
    if len(selected_idx)==now_processed:
        new_params[name]=p1[:,selected_idx[now_processed-1]]
    else:
        new_params[name]=p1[selected_idx[now_processed]][:,selected_idx[now_processed-1]]
else:
    new_params[name]=p1[selected_idx[now_processed]]
:
```

Recurrent Neural Networks for Language Understanding



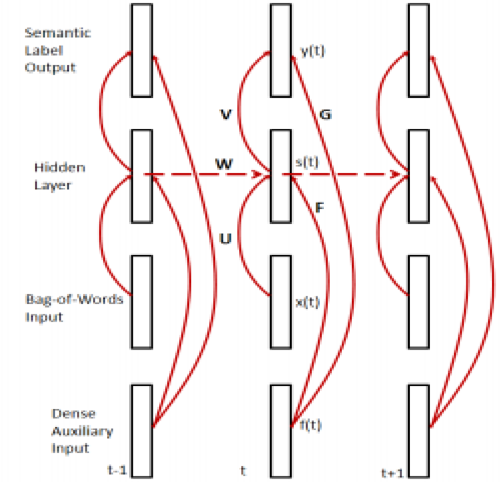
$$\mathbf{s}(t) = f\left(\mathbf{U}\mathbf{w}(t) + \mathbf{W}\mathbf{s}(t-1)\right)$$

$$\mathbf{y}(t) = g\left(\mathbf{V}\mathbf{s}(t)\right),$$

f()为tanh；g()为softmax

1.将input从one-hot变为BoW或n-hot

2.增加未来单词的连续空间矢量，通过上下文窗口滑动来获取位置信息

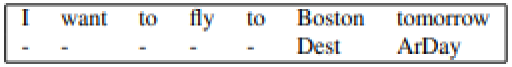


$$\mathbf{s}(t) = f\left(\mathbf{U}\mathbf{x}(t) + \mathbf{W}\mathbf{s}(t-1) + \mathbf{F}\mathbf{f}(t)\right)$$

$$\mathbf{y}(t) = g\left(\mathbf{V}\mathbf{s}(t) + \mathbf{G}\mathbf{f}(t)\right),$$

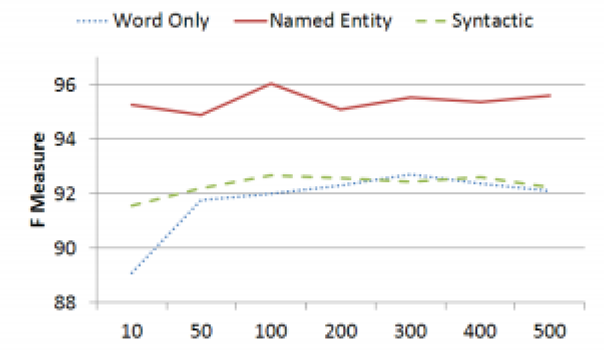
X(t)为2-hot 即{w(t),w(t+1)}
f(t)是由另一个网络学习到的参数

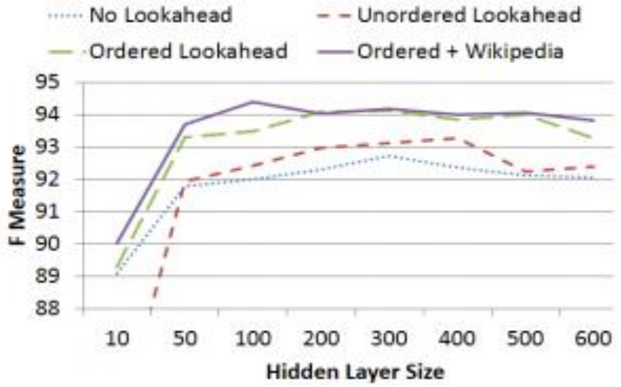
ATIS Dataset

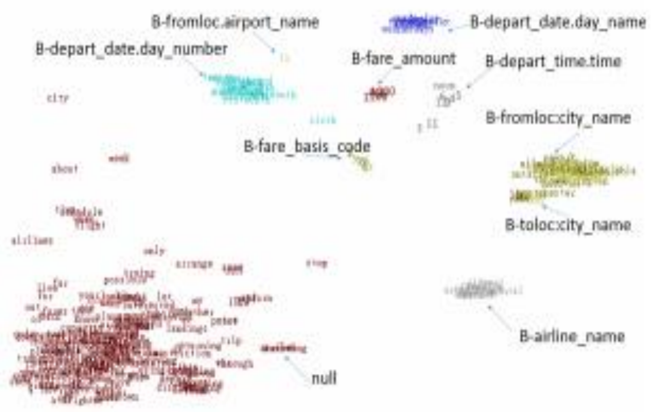


I	want	to	fly	to	Boston	tomorrow
-	-	-	-	-	Dest	ArDay

Named-entity tags:命名标签
eg:”B-city-name” , ”B-time”
Syntactic labels:句法标签
eg:”VBP”,”JJ”


Word Only
Named Entity
Syntactic
F Measure


No Lookahead
Unordered Lookahead
Ordered Lookahead
Ordered + Wikipedia
F Measure
Hidden Layer Size


B-fromloc.airport_name
B-depart_date.day_name
B-depart_date.day_number
B-fare_amount
B-depart_time.time
B-fromloc.city_name
B-fare_basis_code
B-toloc.city_name
B-airline_name
null