MoNet

一个新型神经网络AI表述与实现框架，基于pytorch，scikit-learn，skorch，实现网络的公式化表达与构建。
网络列表如下：

公式	代码	含义	实现
Fc^o_{true}	fc_True	全连接层	`nn.Linear(in_features=i,out_features=o,bias=bias)`
Bfc^o_{true}	bfc_True	双线性层	`nn.Bilinear(in1_features=i[0],in2_features=i[1],out_features=o,bias=b`
$Fl^o_{1,-1}$	flat_1_-1	压扁层	`nn.Flatten(start_dim=start_dim,end_dim=end_dim)`
$Cv2^o_{3,1,0,true}$	cov2_3_1_0_True	卷积层	`eval(f"nn.Conv{dim}d")(in_channels=i,out_channels=o,kernel_size=kerne`
$Cvt2$	covT2_3_1_0_True	反卷积层	`eval(f"nn.ConvTranspose{dim}d") (in_channels=i,out_channels=o,kernel_size=kernel_size,stride=stride,p`
$Mp2$	mp2_2_0	最大池化	`eval(f"nn.MaxPool{dim}d")(kernel_size=kernel_size,padding=padding)`
$Mpa2$	amp2_2_0	自适应最大池化	`eval(f"nn.AdaptiveMaxPool{dim}d")(kernel_size=kernel_size,padding=pad`
$Ap2$	ap2_2	平均池化	`eval(f"nn.AvgPool{dim}d")(padding=padding)`
$Apa2$	aap2_2	自适应平均池化	`eval(f"nn.AdaptiveAvgPool{dim}d")(padding=padding)`

公式	代码	含义	实现
$Bn1$	bn1_0	批归一化层	`eval(f"nn.BatchNorm{dim}d")(num_features=i if num_features==0else num_`
$In1$	in1_0	归一化层	`eval(f"nn.InstanceNorm{dim}d")(num_features=i if num_features==0else`
Gn	gn_2	组归一化层	`nn.GroupNorm(num_groups=num_groups,num_channels=i)`
Ln	ln	归一化层	`nn.LayerNorm(normalized_shape=i)`
Lrn	lrn	归一化层	`nn.LocalResponseNorm(size=i)`
Dp	dp_0.5_False	隐藏层	`nn.Dropout(p=p,inplace=bool(inplace))`
$Dp1$	dp1_1_False	隐藏层	`eval(f"nn.Dropout{dim}d")(p=p,inplace=bool(inplace))`
Adp	aldp_0.5_False	隐藏层	`nn.AlphaDropout(p=p,inplace=bool(inplace))`
$Fadp$	fadp_0.5_False	隐藏层	`nn.FeatureAlphaDropout(p=p,inplace=bool(inplace))`
Act	act.PReLU	激活层	eval(f"nn.{act_func}")()
Nn	nn.Linear_(10,1)	通配层	eval(f"nn.{func}")(*args)

如何用公式表达一个网络

Lenet

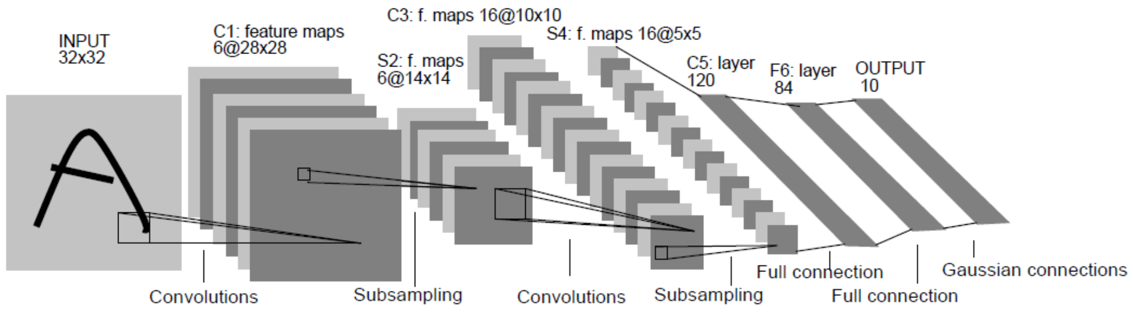

Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

$$Lenet = CvSpCvSpFcFcGc = 2(CvSp)2FcGc$$

$$\begin{aligned} Lenet^{1}_{32\times 32} &= Cv2^{6}_{5}Sp2^{6}_{2}Cv2^{16}_{5}Sp2^{16}_{2}Fc^{120}Fc^{84}Gc^{10} \\ &= (Cv2_5Sp2_2)^{[6,16]}Fc^{[120,64]}Gc^{10} \\ &= (Cv_5Sp_2)2^{[6,16]}Fc^{[120,64]}Gc^{10} \\ &= (Cv_5Sp_2)^{[6,16]}Fc^{[120,64]}Gc^{10} \end{aligned}$$

```
# 代码实现
import monet as mn
Lenet = mn.Mix(1, [[6,16],[120,64],10], [['cv_5','sp_2'],'fc','gc'])
```