SimAM

论文研究内容

论文原文:

“Specifically, we base on some well-known neuroscience theories and propose to optimize an energy function to find the importance of each neuron. We further derive a fast closed-form solution for the energy function, and show that the solution can be implemented in less than ten lines of code.”

总结:

作者基于一些著名的神经科学理论，提出了一个能量函数用来计算feature map中每个神经元的重要性。

原理

论文依据的神经科学理论:

“在视觉神经科学中，信息量最大的神经元通常与周围神经元的发射模式截然不同。此外，活跃的神经元还可能抑制周围神经元的活动，这种现象被称为空间抑制。换句话说，在视觉处理过程中，具有明显空间抑制效应的神经元应该被赋予更高的优先级（即重要性）。”

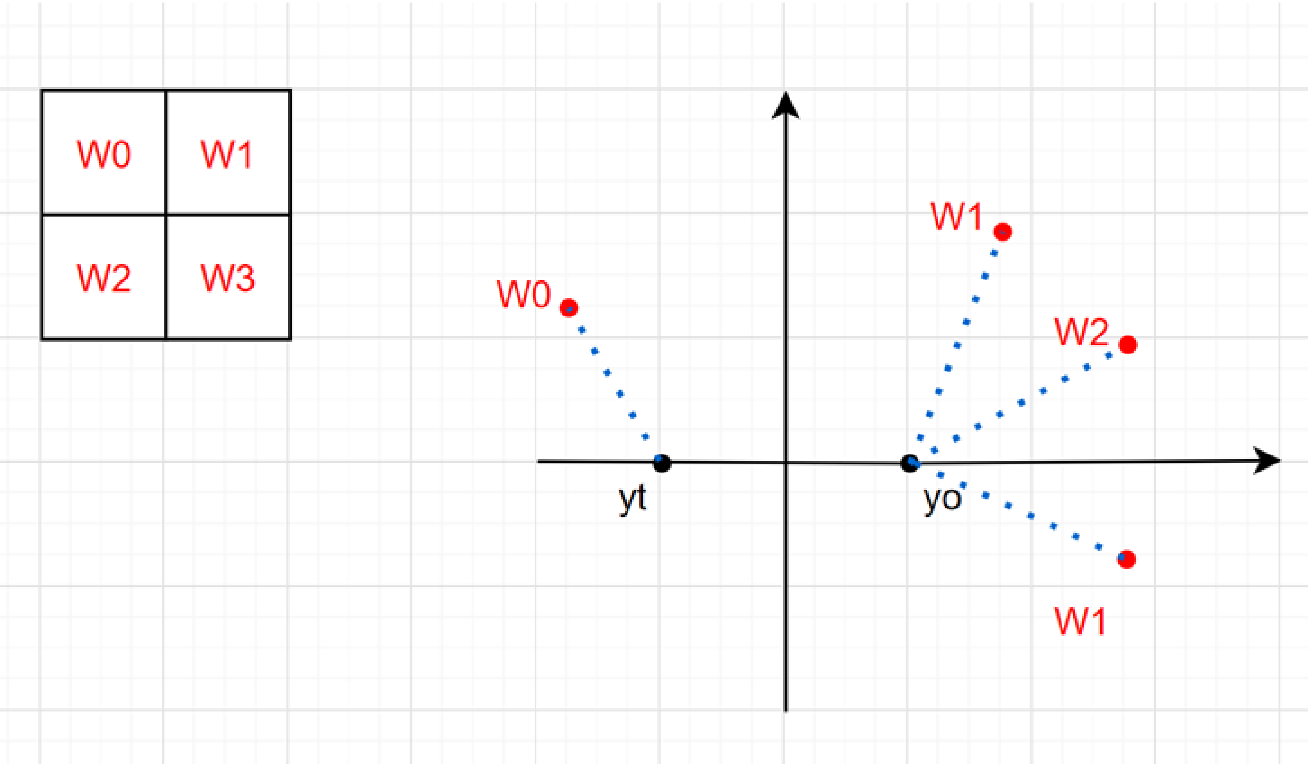
根据以上理论，论文提出了一种方法：测量feature map中的一个目标神经元与其他神经元之间的线性分离度。公式如下:

$$e_t(w_t, b_t, \mathbf{y}, x_i) = (y_t - \hat{t})^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}(y_o - \hat{x}_i)^2. \quad (1)$$

Here, $\hat{t} = w_t t + b_t$ and $\hat{x}_i = w_t x_i + b_t$ are linear transforms of t and x_i, where t and x_i are the target neuron and other neurons in a single channel of the input feature $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$. i is index over spatial dimension and

个人理解：

y_t与y_o是两个对立点， 表示目标神经元在y_t点的离散程度， 表示其他神经元在y_o点的离散程度，用图理解如下：



如此，et越小，则表示目标神经元离yt点越近，其他神经元离yo点越近，最终的结果就是：目标神经元和其他神经元线性分离出来了。
在论文中，为了方便计算，取yt=1，yo=-1，再添加上正则项，最后得到能量函数为：

$$e_t(w_t, b_t, \mathbf{y}, x_i) = \frac{1}{M-1}\sum_{i=1}^{M-1}(-1-(w_t x_i + b_t))^2 + (1-(w_t t + b_t))^2 + \lambda w_t^2. \tag{2}$$

要使et最小，意味着要求导数找极值，但是要对feature map的每一个神经元进行这个过程，计算量是非常大的。由此，论文提出了一个近似解（不大理解怎么推导出来的），如下：

$$w_t = -\frac{2(t-\mu_t)}{(t-\mu_t)^2 + 2\sigma_t^2 + 2\lambda}, \tag{3}$$

$$b_t = -\frac{1}{2}(t+\mu_t)w_t. \tag{4}$$

$\mu_t = \frac{1}{M-1}\sum_{i=1}^{M-1} x_i$ and $\sigma_t^2 = \frac{1}{M-1}\sum_{i}^{M-1}(x_i - \mu_t)^2$ are mean and variance calculated over all neurons except t in that channel. Since existing solutions shown in Eqn (3) and

因此，能量函数的表达式变为：

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t-\hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda}, \tag{5}$$

where $\hat{\mu} = \frac{1}{M}\sum_{i=1}^{M} x_i$ and $\hat{\sigma}^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i - \hat{\mu})^2$. Eqn (5) indicates that the lower energy e_t^*, the neuron t is more distinctive from surround neurons, and more important for visual processing. Therefore, the importance of each neuron can be obtained by $1/e_t^*$. Akin to our method, (Aubry

最后，论文使用1/et作为权重系数，并添加了一个sigmoid函数增加非线性。

$$\widetilde{\mathbf{X}} = sigmoid(\frac{1}{\mathbf{E}}) \odot \mathbf{X}, \tag{6}$$

where $\mathbf{E}$ groups all e_t^* across channel and spatial dimensions. $sigmoid$ is added to restrict too large value in $\mathbf{E}$. It will not influence the relative importance of each neuron because $sigmoid$ is a monofonic function.

pytorch代码

```python
class SimAM(nn.Module):
    def __init__(self, c1, c2):
        super(SimAM, self).__init__()
        self.lamdba = 0.0001
        self.sigmoid = nn.Sigmoid()

    def forward(self, X):
        # spatial size
        n = X.shape[2] * X.shape[3] - 1

        """
        要添加keepdim=True,
        不添加的话，结果的维度会变成2维的，和原来的4维不对应
        例如：原来b*c*h*w ---> b*c
        """
        # square of (t - u)
        x_mean = X.mean(dim=[2, 3], keepdim=True)
        d = (X - x_mean)
        d = d.pow(2)

        """
        要添加keepdim=True,
        不添加的话，结果的维度会变成2维的，和原来的4维不对应
        例如：原来b*c*h*w ---> b*c
        """
        # d.sum() / n is channel variance
        v = d.sum(dim=[2, 3], keepdim=True)
        v = v / n

        # E_inv groups all importance of X
        E_inv = d / (4 * (v + self.lamdba)) + 0.5
```

```
        # return attended features
        return X * self.sigmoid(E_inv)
```