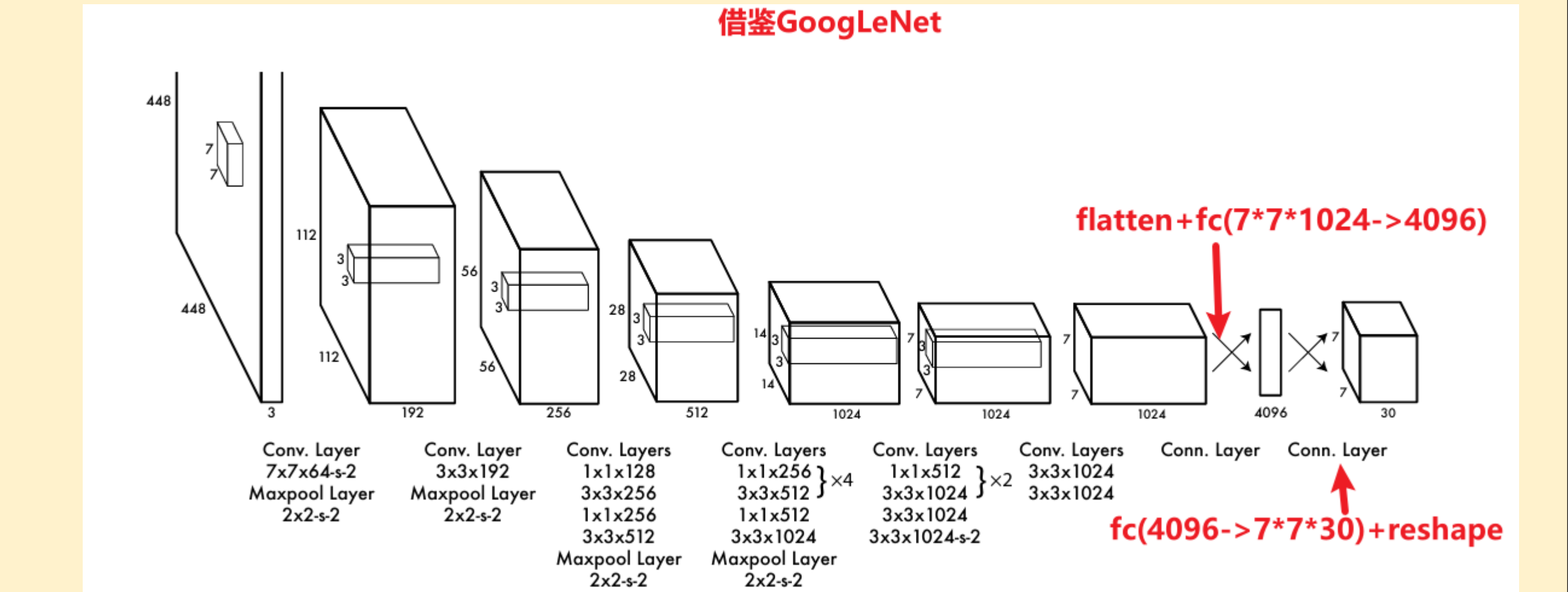
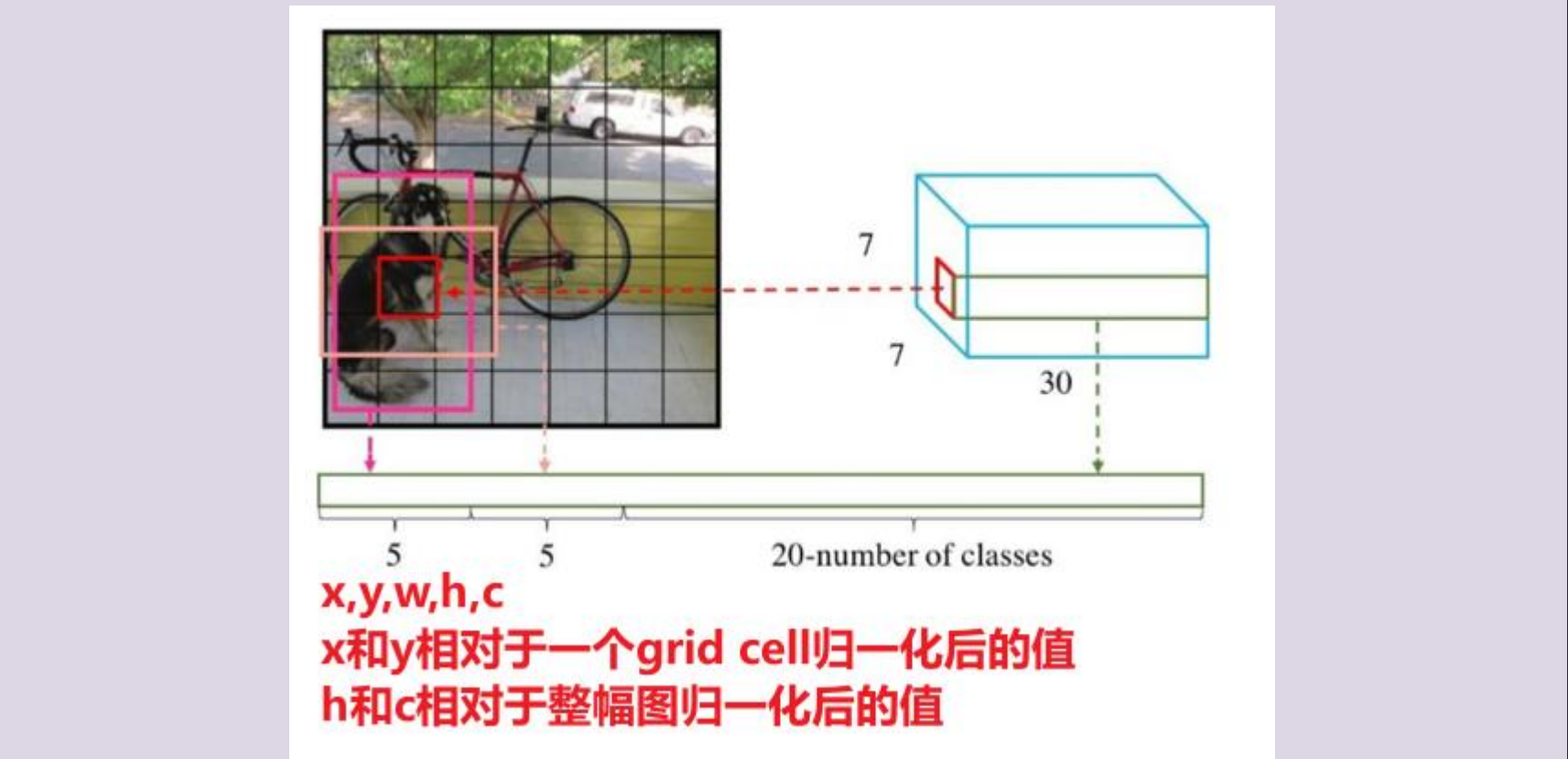


## 网络结构



网络输出



Import torch	损失函数代码
--------------	--------

```

import torch
import torch.nn as nn
class YOLOv1Loss(nn.Module):
    def __init__(self, lambda_coord=5, lambda_noobj=0.5):
        super(YOLOv1Loss, self).__init__()
        self.lambda_coord = lambda_coord
        self.lambda_noobj = lambda_noobj
        self.mse = nn.MSELoss(reduction="sum" )
    def forward(self, pred, target):
        # pred [B, 7, 7, 30]
        # target [B, 7, 7, 30]

```

```
obj_mask = target[... , 4]==1
```

```
# 坐标损失
pred_box = pred[..., 0:4][obj_mask]
target_box = target[..., 0:4][obj_mask]
loss_obj = self.mse(pred_conf_obj, target_conf_obj)
```

```
# 含物体的置信度损失
pred_conf_obj = pred[..., 4][obj_mask]
target_conf_obj = target[..., 4][obj_mask]
loss_obj = self.mse(pred_conf_obj, target_conf_obj)

# 无物体置信度损失
noobj_mask = ~obj_mask
pred_conf_noobj = pred[..., 4][noobj_mask]
target_conf_noobj = target[..., 4][noobj_mask]
loss_noobj = self.lambda_noobj*self.mse(pred_conf_noobj, target_conf_noobj)
```

```
# 分类损失
pred_cls = pred[..., 10:][obj_mask]
target_cls = target[..., 10:][obj_mask]
loss_cls = self.mse(pred_cls, target_cls)
```

```
return loss_coord + loss_obj + loss_noobj + loss_cls
```

## 数据制作

数据集制作成：7\*7\*30

```
例：0, 0.5, 0.5, 0.5, 0.5 # 类别 中心点坐标 宽高
    1, 0.4, 0.6, 0.2, 0.2 # 类别 中心点坐标 宽高
```

1. 生成一个7\*7\*30的全零张量

### 3. 一行一行读取txt文件(下面以第一个举例为例)

4. 计算网格索引  $\text{grid\_x} = \text{int}(x*7)$   $\text{grid\_y} = \text{int}(y*7)$

5. 计算中心点偏移量  $x\_offset = x\_center * 7 - grid\_x$

6. 选择第一个边界框赋值  $\text{tensor}[\text{grid\_y}, \text{grid\_x}, 0:5] = [\text{x\_offset}, \text{y\_offset}, \text{w}, \text{h}, 1]$

7. 类别one-hot编码赋值 `tensor[grid_y,grid_x,10+class_id] = 1`

伪代码:


```
import numpy as np
```

```
Def convert_annotation_to_tensor(annotations,num_class=20):
```

```
tensor = np.zeros((7, 7, 30))
```

 for obj in annotations:

```
class_id, x_center, y_center, w, h = obj
```



```
grid_x = int(x_center*7)
```

```
grid_y = int(y_center*7)
```

made by 迦非赫小堂

```
x_offset = x_center*7 - grid_x
```

```
y_offset = y_center*7 - grid_y
```

```
tensor[grid_y, grid_x, 0:5] = [x_offset, y_offset, w, h, 1]
```

```
tensor[grid_y, grid_x, 10+class_id] = 1
```

```
return torch.from_numpy(tensor)
```

## 损失函数公式

**坐标损失数学表达式：均方误差损失MSE**

$$L_{\text{coord}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{I}_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

### 置信度损失数学表达式

$$L_{\text{conf,obj}} = \sum_{i=1}^N 1_{\text{obj}} (C_{\text{pred}} - C_{\text{true}})^2$$

$$L_{\text{conf, noobj}} = \sum_{i=1}^N 1_{\text{noobj}} \lambda_{\text{noobj}} (C_{\text{pred}})^2$$

$$L_{\text{conf}} = L_{\text{conf,obj}} + L_{\text{conf,noobj}}$$

类别损失数学表达式

$$L_{\text{cls}} = \sum_{i=1}^N \sum_{c \in \text{classes}} (p_{\text{pred}}(c) - p_{\text{true}}(c))^2$$

网格单元总数 类别数量  
有无数物体