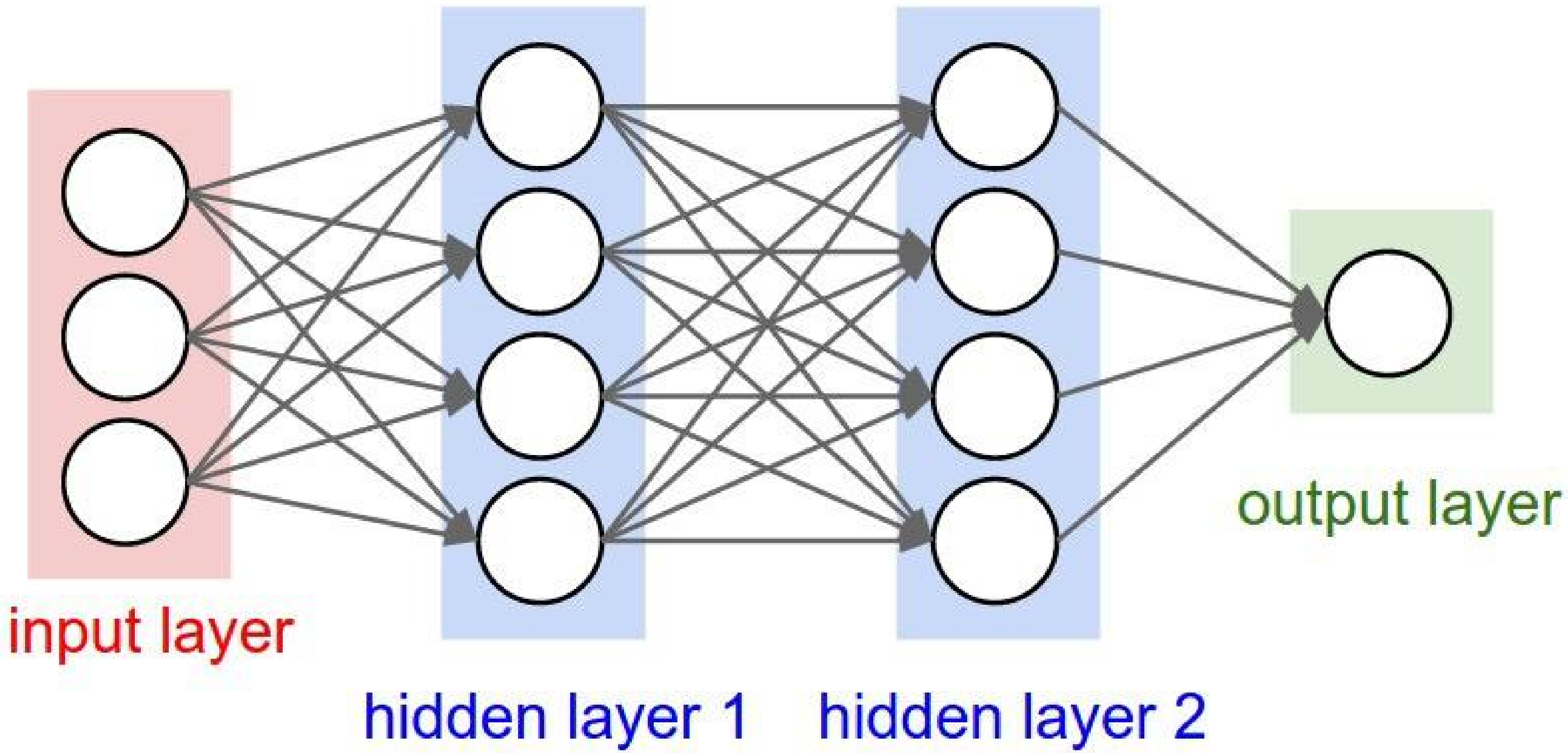


深度学习框架Caffe学习与应用 第8课

DATAGURU专业数据分析社区

本节课内容

- 动手编写CNN框架（初级）
- 1. 矩阵运算
- 2. Caffe的最小可用版本

矩阵运算

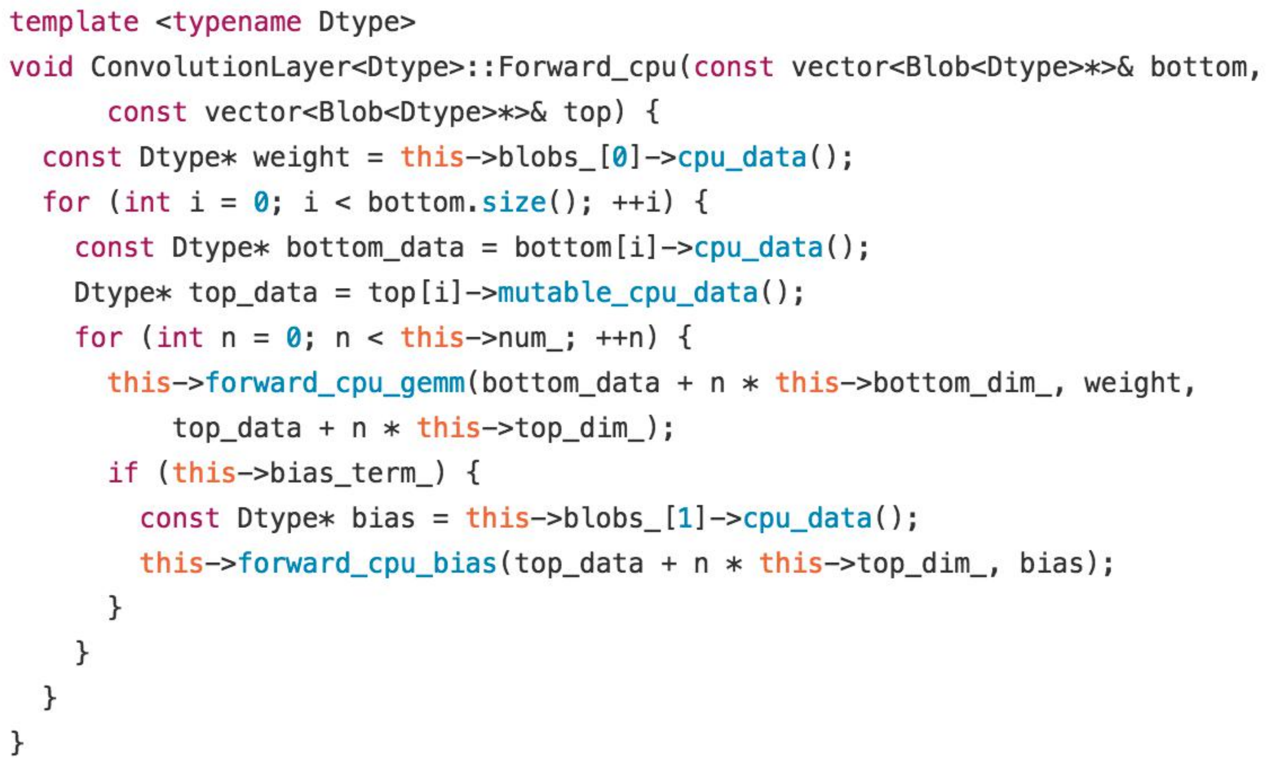
```
void cblas_gemm (
const enum CBLAS_ORDER Order,  // Specifies row-major (C) or column-major (Fortran) data ordering.
const enum CBLAS_TRANSPOSE TransA,//Specifies whether to transpose matrix A.
const enum CBLAS_TRANSPOSE TransB,
const int M,   //Number of rows in matrices A and C.
const int N,//Number of rows in matrices A and C.
const int K,  //Number of columns in matrix A; number of rows in matrix B
const float alpha, //Scaling factor for the product of matrices A and B
const float *A,
const int lda, //The size of the first dimention of matrix A; if you are passing a matrix A[m][n], the value
should be m.  stride
const float *B,
const int ldb,  //The size of the first dimention of matrix B; if you are passing a matrix B[m][n], the value
should be m.
const float beta,  //Scaling factor for matrix C.
float *C,
const int ldc    //The size of the first dimention of matrix C; if you are passing a matrix C[m][n], the value
should be m.
);
```

- 1.CBLAS_ORDER解释：
- Matrix A =
- [1 2 3]
- [4 5 6]
- Row-major stores values as {1,2,3,4,5,6}
- Col-major stores values as {1, 4, 2, 5, 3, 6}

- 2. 函数计算内容：
- C←αAB + βC

- Caffe中使用了cblas_gemm的使用例子：
- 1. caffe_cpu_gemm()
- 2. forward_cpu_gemm()
- 3. forward_cpu_bias()
- …

- https://github.com/BVLC/caffe/blob/master/src/caffe/layers/conv_layer.cpp



```
template <typename Dtype>
void ConvolutionLayer<Dtype>::Forward_cpu(const vector<Blob<Dtype>*>& bottom,
      const vector<Blob<Dtype>*>& top) {
  const Dtype* weight = this->blobs_[0]->cpu_data();
  for (int i = 0; i < bottom.size(); ++i) {
    const Dtype* bottom_data = bottom[i]->cpu_data();
    Dtype* top_data = top[i]->mutable_cpu_data();
    for (int n = 0; n < this->num_; ++n) {
      this->forward_cpu_gemm(bottom_data + n * this->bottom_dim_, weight,
          top_data + n * this->top_dim_);
      if (this->bias_term_) {
        const Dtype* bias = this->blobs_[1]->cpu_data();
        this->forward_cpu_bias(top_data + n * this->top_dim_, bias);
      }
    }
  }
}
```

Caffe最小版本

- 数据结构Blob：
- struct Blob{
- vector<int> shape_; // Blob的形状[num,channel,width,height]
- float* data_; // 数据,向前传播使用
- float* diff_; // 梯度，先后传播使用
- };

- 层Layer：
- class Layer{
- public:
-   void forward(vector<Blob>* bottom, vector<Blob>* top);
-   void backward(vector<Blob>* bottom, vector<Blob>* top);
- protected:
-   vector<Blob> blob_; // 存放学习参数，如weight,bias
- }

- 以卷积层、池化层、全连接层和激活函数层为例回看Caffe源码。