



- 博客 (http://blog.csdn.net/?ref=toolbar)
- 学院 (http://edu.csdn.net/?ref=toolbar)
- 下载 (http://download.csdn.net/?ref=toolbar)
- GitChat (http://gitbook.cn/?ref=csdn) 更多 ▾
- 0

Q

🔍

📁
- 登录 (https://passport.csdn.net/account/login?ref=toolbar)
- 注册 (https://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister)
- caffe中各种cblas的函数使用总结

转载

2017年04月18日 14:39:06

502

转Ⓜ: <http://www.cnblogs.com/huashiyiqike/p/3886670.html> 和 http://blog.sina.com.cn/s/blog_4a03c0100101ethf.html

```
Y=alpha * X +beta*Y
template <>
void caffe_cpu_axpby<float>(const int N, const float alpha, const float* X,
                           const float beta, float* Y) {
    cblas_saxpby(N, alpha, X, 1, beta, Y, 1);
}

template <>
void caffe_cpu_axpby<double>(const int N, const double alpha, const double* X,
                             const double beta, double* Y) {
    cblas_daxpby(N, alpha, X, 1, beta, Y, 1);
}

cblas_dscal(N, beta, Y, incY); Y=Y*beta
cblas_daxpy(N, alpha, X, incX, Y, incY); Y= (alpha * X) + Y

Y=alpha * X + Y
template <>
void caffe_axpy<float>(const int N, const float alpha, const float* X,
                      float* Y) { cblas_saxpy(N, alpha, X, 1, Y, 1); }

template <>
void caffe_axpy<double>(const int N, const double alpha, const double* X,
                       double* Y) { cblas_daxpy(N, alpha, X, 1, Y, 1); }

DEFINE_VSL_BINARY_FUNC(Add, y[i] = a[i] + b[i]);
DEFINE_VSL_BINARY_FUNC(Sub, y[i] = a[i] - b[i]);
DEFINE_VSL_BINARY_FUNC(Mul, y[i] = a[i] * b[i]);
DEFINE_VSL_BINARY_FUNC(Div, y[i] = a[i] / b[i]);

template <>
void caffe_add<float>(const int n, const float* a, const float* b,
                    float* y) {
    vsAdd(n, a, b, y);
}

template <>
void caffe_add<double>(const int n, const double* a, const double* b,
                      double* y) {
    vdAdd(n, a, b, y);
}

y=x;
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



时光碎了天 (http://blog....)

+关注

(http://blog.csdn.net/u013289254)

原创	粉丝	喜欢	码云未开通
72	23	0	(https://gitee.com/u013289254)

他的最新文章
更多文章 (http://blog.csdn.net/u013289254)

【git使用】Permission denied (publickey). fatal: Could not read from remote repository.解决办法 (http://blog.csdn.net/u013289254/article/details/78527584)

机器学习中的特征选择 (http://blog.csdn.net/u013289254/article/details/77991829)

动态规划总结 (http://blog.csdn.net/u013289254/article/details/77865212)

大数据-总结 (http://blog.csdn.net/u013289254/article/details/77849020)

GBDT (MART) 迭代决策树入门教程 | 简介 (http://blog.csdn.net/u013289254/article/details/77678465)



Unable to Conn

The Proxy was unable to connect to the remote site. responding to requests. If you feel you have reached please submit a ticket via the link provided below.

URL: <http://pos.baidu.com/s?hei=250&wid=300&di=u%2Fblog.csdn.net%2Fu013289254%2Farticle%2Fdetails/70438047>

他的热门文章

使用caffe训练的深度学习做目标检测(车辆检测) (http://blog.csdn.net/u013289254/article/details/70438047)

2468 登录 注册 X

```

template <>
void caffe_copy<float>(const int N, const float* X, float* Y) {
    cblas_scopy(N, X, 1, Y, 1);
}

template <>
void caffe_copy<double>(const int N, const double* X, double* Y) {
    cblas_dcopy(N, X, 1, Y, 1);
}

template <>
void caffe_gpu_copy<float>(const int N, const float* X, float* Y) {
    CUBLAS_CHECK(cublasScopy(Caffe::cublas_handle(), N, X, 1, Y, 1));
}

template <>
void caffe_gpu_copy<double>(const int N, const double* X, double* Y) {
    CUBLAS_CHECK(cublasDcopy(Caffe::cublas_handle(), N, X, 1, Y, 1));
}

```

Computes $\alpha x'y' + A$.

`cblas_sger`

Multiplies vector X by the transform of vector Y, then adds matrix A (single precision).

Multiplies vector X by the transform of vector Y, then adds matrix A (single precision).

```

void cblas_sger (
    const enum CBLAS_ORDER Order,
    const int M,
    const int N,
    const float alpha,
    const float *X,
    const int incX,
    const float *Y,
    const int incY,
    float *A,
    const int lda
);

```

$Y(\text{vector}) \leftarrow \alpha AX + \beta Y$

This function multiplies $A * X$ (after transposing A, if needed) and multiplies the resulting matrix by alpha. It then multiplies vector Y by beta. It stores the sum of these two products in vector Y.

```

template <>
void caffe_cpu_gemv<float>(const CBLAS_TRANSPOSE TransA, const int M,
    const int N, const float alpha, const float* A, const float* x,
    const float beta, float* y) {
    cblas_sgemv(CblasRowMajor, TransA, M, N, alpha, A, N, x, 1, beta, y, 1);
}

```

$C(\text{matrix}) \leftarrow \alpha AB + \beta C$

```

template<typename T>
void gpu_multmat(T* A, T* B, T* C, int M,int K,int N){
    const T alpha = 1,beta=0;
    caffe_gpu_gemm(CblasNoTrans,CblasNoTrans,M,N,K,alpha,A,B,beta,C);
}

```

```

template<>
void caffe_cpu_gemm<float>(const CBLAS_TRANSPOSE TransA,
    const CBLAS_TRANSPOSE TransB, const int M, const int N, const int K,
    const float alpha, const float* A, const float* B, const float beta,
    float* C) {
    int lda = (TransA == CblasNoTrans) ? K : M;
    int ldb = (TransB == CblasNoTrans) ? N : K;
    cblas_sgemm(CblasRowMajor, TransA, TransB, M, N, K, alpha, A, lda, B,
        ldb, beta, C, N);
}

```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

行人检测数据集 (<http://blog.csdn.net/u013289254/article/details/70337121>)

1900

深入理解K-Means聚类算法 (<http://blog.csdn.net/u013289254/article/details/59108855>)

1128

(16)caffe总结之自定义数据输入层 (<http://blog.csdn.net/u013289254/article/details/70432304>)

1093

矩阵求导公式 (<http://blog.csdn.net/u013289254/article/details/69055876>)

969

登录

注册

×

```

A=M*N  B=M*K
C=A*B   N M K

template<typename T>
void cpu_multTmat(T* A, T* B, T* C, int M,int K,int N){
    const T alpha = 1,beta=0;
    caffe_cpu_gemm(CblasTrans,CblasNoTrans,M,N,K,alpha,A,B,beta,C);
    // cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, M, N, K, alpha, A, M, B, K, beta, C, M);
}

A=M*N B=N*K
C=A*B   M N K

template<typename T>
void cpu_multmat(T* A, T* B, T* C, int M,int K,int N){
    const T alpha = 1,beta=0;
    caffe_cpu_gemm(CblasNoTrans,CblasNoTrans,M,N,K,alpha,A,B,beta,C);
    // cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, M, N, K, alpha, A, M, B, K, beta, C, M);
}

```

计算矩阵乘法的函数之一是 `cblas_sgemm`，使用单精度实数，另外还有对应双精度实数，单精度复数和双精度复数的函数。在此以 `cblas_sgemm` 为例。

函数定义为：

```

void cblas_sgemm(const enum CBLAS_ORDER Order, const enum CBLAS_TRANSPOSE TransA,
const enum CBLAS_TRANSPOSE TransB, const int M, const int N,
const int K, const float alpha, const float *A,
const int lda, const float *B, const int ldb,
const float beta, float *C, const int ldc)

```

得到的结果是：

$C = \alpha * op(A) * op(B) + \beta * C$

`const enum CBLAS_ORDER Order`，这是指的数据的存储形式，在CBLAS的函数中无论一维还是二维数据都是用一维数组存储，这就要涉及是行主序还是列主序，在C语言中数组是用行主序，fortran中是列主序。我还是习惯于是用行主序，所以这个参数是用 `CblasRowMajor`，如果是列主序的话就是 `CblasColMajor`。

`const int M`，矩阵A的行，矩阵C的行

`const int N`，矩阵B的列，矩阵C的列

`const int K`，矩阵A的列，矩阵B的行

`const float alpha`，`const float beta`，计算公式中的两个参数值，如果只是计算 $C=A*B$ ，则 $\alpha=1, \beta=0$

`const float *A`，`const float *B`，`const float *C`，矩阵ABC的数据

`const int lda`，`const int ldb`，`const int ldc`，在BLAS的文档里，这三个参数分别为ABC的行数，但是实际使用发现，在CBLAS里应该是列数。



caffe中各种cblas的函数使用总结 (<http://blog.csdn.net/g110802008/article/details/53434...>)

总结的很好，好东西分享，尊重原始作者Here $Y = \alpha * X + \beta * Y$ $Y = \alpha * X + \beta * Y$ template void caffe_cpu_axpby(const int



g110802008 (<http://blog.csdn.net/g110802008>) 2016年12月02日 14:26 980

caffe常用的cblas函数 (<http://blog.csdn.net/zyf19930610/article/details/71404814>)

转自 <http://www.cnblogs.com/huashiyiqike/p/3886670.html> $Y = \alpha * X + \beta * Y$ template v...



zyf19930610 (<http://blog.csdn.net/zyf19930610>) 2017年05月08日 11:43 135

对函数 cblas_sgemm的说明 (<http://blog.csdn.net/zhuiquik/article/details/72916558>)

计算矩阵乘法的函数之一是 cblas_sgemm，使用单精度实数，另外还有对应双精度实数，单精度复数和双精度复数的函数。在此以 cblas_sgemm为例。函数定义为： void c...



zhuiquik (<http://blog.csdn.net/zhuiquik>) 2017年06月08日 14:08 254

cblas_cgemm()函数说明 (<http://blog.csdn.net/u011283536/article/details/74216485>)

楼主最近在用blas库做矩阵的运算，用到了复数矩阵的乘法，然后去查cblas_cgemm()函数，可是转了一圈，没发现这个函数的举例说明，大都是cblas_sgemm()和cblas_dgemm()的...



u011283536 (<http://blog.csdn.net/u011283536>) 2017年07月03日 16:45 114

CBLAS编译安装与使用举例 (<http://blog.csdn.net/zouyu1746430162/article/details/5339...>)

在Github上看到有人用BLAS library优化自己的源码，对此产生了强烈兴趣。准备自己动手实践一下，网上搜索了一大堆编译安装BLAS教程的资料，没一个靠谱的，编译过程中遇到一堆的问题。...



zouyu1746430162 (<http://blog.csdn.net/zouyu1746430162>) 2016年11月29日 11:50 541

CBLAS编译安装与使用 (<http://blog.csdn.net/zuoyigexingfude/article/details/50488255>)

在Github上看到有人用BLAS library优化自己的源码，对此产生了强烈兴趣。准备自己动手实践一下，网上搜索了一大堆编译安装BLAS教程的资料，没一个靠谱的，编译过程中遇到一堆的问题。因为...



zuoyigexingfude (<http://blog.csdn.net/zuoyigexingfude>) 2016年01月09日 15:58 778



cblas代码，可直接使用 (http://download.csdn.net/download/weixin_38...)

http://download.csdn.net/download/weixin_38...

2017年05月09日 02:27 193KB

下载



CBLAS的安装与使用 (<http://download.csdn.net/download/u011768043/...>)

<http://download.csdn.net/download/u011768043/...>

2013年08月21日 13:23 43KB

下载

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！


登录

注册



使用caffe自带的工具绘制loss函数曲线 ([http://blog.csdn.net/TracelessLe/article/details/...](http://blog.csdn.net/TracelessLe/article/details/)

由于最近使用caffe训练一个网络，学习了如何使用训练过程中生成的log来绘制loss函数曲线，主要用于观察所训练的网络是否收敛。前提：安装完caffe后已经对pycaffe和matcaffe编译...

 TracelessLe (<http://blog.csdn.net/TracelessLe>) 2017年05月05日 21:22  295



caffe 加权交叉熵损失函数 SigmoidCrossEntropyLoss ([http://download...](http://download.csdn.net/download/TracelessLe/9669338)

<http://download.csdn.net/download/TracelessLe/9669338> 2017年12月24日 23:13 4KB [下载\(\)](#)





caffe中python接口的安装和使用 (<http://download.csdn.net/download/huizhuangzhuoping/article/details/50756141>

<http://download.csdn.net/download/huizhuangzhuoping/article/details/50756141> 2016年01月15日 22:47 2KB [下载\(\)](#)



走进blas/lapack (3) --cblas (<http://blog.csdn.net/zhf2016/article/details/52590654>)

cblas cblas是一个C语言的接口，更方便用C来调用，而不用与fortran混合编程。因此，输入的变量不必是指针，函数名不用加上下划线，等等。下载的地址就在blas的地址blas上。 cbl...

 zhf2016 (<http://blog.csdn.net/zhf2016>) 2016年09月19日 23:40  615





深度学习框架技术解析及caffe实战总结 ([http://download.csdn.net/dow...](http://download.csdn.net/download/TracelessLe/9669338)

<http://download.csdn.net/download/TracelessLe/9669338> 2017年12月29日 19:02 2.52MB [下载\(\)](#)

使用Caffe对图像做训练、分类的流程总结 ([http://blog.csdn.net/zizi7/article/details/5073...](http://blog.csdn.net/zizi7/article/details/5073344)

1. 环境操作系统：windows7，显卡：NVIDIA GTX 780 Caffe是直接来自官方GitHub上down下来的，之前安装教程推荐了H appynear的版本，但不知道为什么，我只要...

 zizi7 (<http://blog.csdn.net/zizi7>) 2016年02月24日 17:03  5130



Caffe接口使用模板 (<http://download.csdn.net/download/khtao/9669338>)

<http://download.csdn.net/download/khtao/9669338> 2016年10月31日 20:23 6KB [下载\(\)](#)





caffe-master(要使用cudnn-v3.0,cuda7.0) ([http://download.csdn.net/d...](http://download.csdn.net/download/TracelessLe/9669338)

<http://download.csdn.net/download/TracelessLe/9669338> 2016年09月03日 10:03 7.13MB [下载\(\)](#)

使用caffe训练faster-rcnn时遇到的问题总结 ([http://blog.csdn.net/u012283902/article/det...](http://blog.csdn.net/u012283902/article/details/)

经过今天一天的实验，充分了解了在深度学习领域，显存大的重要性。不得不承认，有了N卡以后，效率提高了很多，当然这也得益于纠缠于cpu训练faster rcnn的日子.... (1) 今天早上，主要解...



 u012283902 (<http://blog.csdn.net/u012283902>) 2017年03月18日 16:41  5325



Atlas lapack cblas for windows x86 and x64 ([http://download.csdn.net/...](http://download.csdn.net/download/TracelessLe/9669338)



<http://download.csdn.net/download/TracelessLe/9669338> 2014年09月08日 17:21 6.15MB [下载\(\)](#)

```
#include #include extern "C" { #include }using namespace std;int main(int argc, char const *argv[...
```

 iamzhangzhuping (<http://blog.csdn.net/iamzhangzhuping>) 2016年02月27日 19:44  1191

激活函数与caffe及参数 (<http://blog.csdn.net/fabulousli/article/details/52890463>)

为什么引入非线性激励函数？如果不用激励函数（其实相当于激励函数是 $f(x) = x$ ），在这种情况下你每一层输出都是上层输入的线性函数，很容易验证，无论你神经网络有多少层，输出都是输入的线性组合，与没...

 fabulousli (<http://blog.csdn.net/fabulousli>) 2016年10月22日 10:44  1293

