ML Operators

BMLS Team

October 26, 2023

Contents

Chapter 1

Introduction to BMLS

Basic Machine Learning Subsystems aims to provide highly performant implementations of common operations in machine learning *and* their gradients. As the complexity and quantity of operators increases, BMLS provides a way to manage that complexity. We do this by formally defining the *forward* and *backward* pass of each operator, peer reviewing those definitions, and providing extensive unit testing. This way, implementers can be absolutely certain the operation is correct.

BMLS is *not* a machine-learning library. It is a *math* library on which machine learning libraries, such as for reverse-mode automatic differentiation, can be implemented.

Data Formats All Tensors in BMLS are assumed to be row-major, NCHW tensors where N = batch size, C = channels, H = height, and W = width. Rank 2 Tensors are assumed to be NC, Rank 3 are assumed to be CHW, and Rank 4 are assumed to be NCHW.

Design Patterns All operations provided by BMLS have a forward operation `name()` and a backwards operation `name_wrt_xn()` for each differentiable input. Operation inputs are `x1`, `x2`, `x3`, etc, followed by an output `y` and gradients are denoted `g1`, `g2`, etc. Inputs are always `*const` and outputs are always `*mut`.

Features BMLS provides element wise operators, activation functions, higher rank operators, loss functions, and gradients for all. It does not, however, implement any error handling. You are expected to implement error handling on an as-needed basis.

1.1 Contributors

- Rylan W. Yancey

1.2 Peer Reviewers

- J. Leon Ballentine

Chapter 2

Element-Wise Operators

2.1 Add

The Addition Operator is defined as $f(u, v) = u + v$. To find the gradient w.r.t u and v, we will use the sum rule, which is defined as:

$$\frac{\delta}{\delta x}(u + v) = \frac{\delta u}{\delta x} + \frac{\delta v}{\delta x}$$

To find the gradient w.r.t u, we will set u as our x and treat v as a constant, which gives us the following:

$$\frac{\delta}{\delta u}(u + v) = \frac{\delta u}{\delta u} + \frac{\delta v}{\delta u} = 1$$

To find the gradient w.r.t v, we will set v as our x and treat u as a constant, which gives us the following:

$$\frac{\delta}{\delta v}(u + v) = \frac{\delta u}{\delta v} + \frac{\delta v}{\delta v} = 1$$

Therefore, we can say that the gradient w.r.t u is 1, and the gradient w.r.t v is 1.

2.2 Sub

The Subtraction Operator is defined as $f(u, v) = u - v$. To find the gradient w.r.t u and v, we will use the subtraction rule, which is defined as:

$$\frac{\delta}{\delta x}(u - v) = \frac{\delta u}{\delta x} - \frac{\delta v}{\delta x}$$

To find the gradient w.r.t u, we will set u as our x and treat v as a constant, which gives us the following:

$$\frac{\delta}{\delta u}(u-v) = \frac{\delta u}{\delta u} - \frac{\delta v}{\delta u} = 1$$

To find the gradient w.r.t v, we will set v as our x and treat u as a constant, which gives us the following:

$$\frac{\delta}{\delta v}(u-v) = \frac{\delta u}{\delta v} - \frac{\delta v}{\delta v} = -1$$

Therefore, we can say that the gradient w.r.t u is 1, and the gradient w.r.t v is -1.

2.3 Mul

The Multiplication Operator is defined as $f(u,v) = uv$. To find the gradient w.r.t u and v, we will use the product rule of derivatives, which is defined as:

$$\frac{\delta}{\delta x}(uv) = v\frac{\delta u}{\delta x} + u\frac{\delta v}{\delta x}$$

To find the gradient w.r.t u, we will set u as our x and treat v as a constant, which gives us the following:

$$\frac{\delta}{\delta u}(uv) = v\frac{\delta u}{\delta u} + u\frac{\delta v}{\delta u} = v$$

To find the gradient w.r.t v, simply do the same, this time setting v as x and treat u as constant.

$$\frac{\delta}{\delta v}(uv) = v\frac{\delta u}{\delta v} + u\frac{\delta v}{\delta v} = u$$

Therefore, we can say that the gradient w.r.t u is v, and the gradient w.r.t v is u.

2.4 Div

The Division Operator is defined as $f(u,v) = \frac{u}{v}$. To find the gradient w.r.t x and y, we will use the quotient rule of derivatives, which is defined as:

$$\frac{\delta}{\delta x}(\frac{u}{v}) = \frac{(u\frac{\delta}{\delta x})v - u(v\frac{\delta}{\delta x})}{v^2}$$

To find the gradient w.r.t u, we will set u as our x and treat v as constant, which gives us the following:

$$\frac{\delta}{\delta u}(\frac{u}{v}) = \frac{v\frac{\delta u}{\delta u} - u\frac{\delta v}{\delta u}}{v^2} = \frac{v - u\frac{0}{\delta u}}{v^2} = \frac{v}{v^2} = \frac{1}{v}$$

To find the gradient w.r.t v, we will set v as our x and treat u as constant, which gives us the following:

$$\frac{\delta}{\delta v}(\frac{u}{v}) = \frac{v\frac{\delta u}{\delta v} - u\frac{\delta v}{\delta v}}{v^2} = \frac{v\frac{0}{\delta v} - u}{v^2} = -\frac{u}{v^2}$$

Therefore, we can say that the gradient w.r.t u is $\frac{1}{v}$, and the gradient w.r.t v is $-\frac{u}{v^2}$.

2.5 Axis Add

The Axis Add Operator iterates over an axis of input A and adds the B value corresponding that that index of the axis. B is a vector that has the same length as the axis to be added to, and A is a tensor. In terms of B_i over axis A_i, where B_i is a value and A_i is a vector of length n.

$$f(A_i, B_i) = \sum_{j=0}^{n} Cij = A_{ij} + B_i$$

Gradient By definition, the axis add operator is a kind of element-wise addition. To find the gradient w.r.t. B_i, we can apply the sum rule.

$$\frac{\delta}{\delta B_i}(\sum_{j=0}^{n} A_{ij} += B_i) = \sum_{i=0}^{n} 1 = n$$

Therefore, we can say the gradient w.r.t. B_i is the sum the gradients of all its additions. The gradient w.r.t. A is 1, since it is a simple element-wise addition.

We must apply the chain rule to this operation to backpropogate with it. To do so, we will multiply the gradient w.r.t. output C $\frac{\delta L}{\delta C_i}$.

$$\frac{\delta L}{\delta B_i} = \sum_{i=0}^{n} \frac{\delta L}{\delta C_i}$$

2.6 Axis Sub

The Axis Add Operator iterates over an axis of input A and subtracts the B value corresponding that that index of the axis. B is a vector that has the same length as the axis to be added to, and A is a tensor. In terms of B_i over axis A_i, where B_i is a value and A_i is a vector of length n.

$$f(A_i, B_i) = \sum_{j=0}^{n} C_{ij} = A_{ij} - B_i$$

Gradient By definition, the axis sub operator is a kind of element-wise subtraction. To find the gradient w.r.t. B_i, we can apply the subtraction rule.

$$\frac{\delta}{\delta B_i}(\sum_{j=0}^{n} C_{ij} = A_{ij} - B_i) = \sum_{i=0}^{n} -1 = n$$

Therefore, we can say the gradient w.r.t. B_i is the sum the gradients of all its subtractions. The gradient w.r.t. A is 1, since it is a simple element-wise subtraction.

We must apply the chain rule to this operation to backpropogate with it. To do so, we will multiply the gradient w.r.t. output C $\frac{\delta L}{\delta C_i}$.

$$\frac{\delta L}{\delta B_i} = \sum_{i=0}^{n} -\frac{\delta L}{\delta C_i}$$

2.7 Axis Mul

The Axis Mul Operator iterates over an axis of input A and multiplies the B value corresponding that that index of the axis. B is a vector that has the same length as the axis to be added to, and A is a tensor. In terms of B_i over axis A_i, where B_i is a value and A_i is a vector of length n.

$$f(A_i, B_i) = \sum_{j=0}^{n} C_{ij} = A_{ij} B_i$$

Gradient w.r.t. B By definition, the axis mul operator is a kind of element-wise multiplication. To find the gradient w.r.t. B_i, we can apply the product rule.

$$\frac{\delta}{\delta B_i}(\sum_{j=0}^{n} C_{ij} = A_{ij} B_i) = \sum_{i=0}^{n} A_{ij}$$

Therefore, we can say the gradient w.r.t. B_i is the sum the gradients of all its multiplications.additions.

We must apply the chain rule to this operation to backpropogate with it. To do so, we will multiply the gradient w.r.t. output C $\frac{\delta L}{\delta C_i}$.

$$\frac{\delta L}{\delta B_i} = \sum_{i=0}^{n} \frac{\delta L}{\delta C_i} A_{ij}$$

Gradient w.r.t. A Applying the same logic, we can conclude that the gradient w.r.t. A_{ij} is the value in B it was multiplied by.

$$\frac{\delta}{\delta A_{ij}} = (A_{ij} B_i) = B_i$$

2.8 Axis Div

The Axis Div Operator iterates over an axis of input A and multiplies the B value corresponding that that index of the axis. B is a vector that has the same length as the axis to be added to, and A is a tensor. In terms of B_i over axis A_i, where B_i is a value and A_i is a vector of length n.

$$f(A_i, B_i) = \sum_{j=0}^{n} C_{ij} = \frac{A_{ij}}{B_i}$$

Gradient w.r.t. B By definition, the axis div operator is a kind of element-wise division. To find the gradient w.r.t. B_i, we can apply the division rule.

$$\frac{\delta}{\delta B_i}(\sum_{j=0}^{n} C_{ij} = \frac{A_{ij}}{B_i}) = \sum_{i=0}^{n} \frac{A_{ij}}{B_i^2}$$

Therefore, we can say the gradient w.r.t. B_i is the sum the gradients of all its divisions.

We must apply the chain rule to this operation to backpropogate with it. To do so, we will multiply the gradient w.r.t. output C_i $\frac{\delta L}{\delta C_i}$.

$$\frac{\delta L}{\delta B_i} = \sum_{i=0}^{n} \frac{\delta L}{\delta C_i} \frac{A_{ij}}{B_i^2}$$

Gradient w.r.t. A Applying the same logic, we can conclude that the gradient w.r.t. A_{ij} is the value in B it was multiplied by.

$$\frac{\delta}{\delta A_{ij}}(\frac{A_{ij}}{B_i}) = \frac{1}{B_i}$$

2.9 Dropout

The Dropout Operator is used to prevent overfitting. It does this by randomly selecting α percentage of values to set to zero. α is a hyperparameter that can be tuned, but is usually set between 0.5 and 0.2. In pure math, with r representing a random value between 0 and 1,

$$\begin{cases} 0 & r < \alpha \\ x\frac{1}{1-\alpha} & r \geq \alpha \end{cases}$$

When $r > \alpha$, we scale x up by a factor of $x\frac{1}{1-\alpha}$. In practice, this prevents neurons from finding relationships between neurons that are not significant, which tends to happen when networks are larger than they need to be or there is insufficient data. During test or inference, the dropout layer is turned off.

Gradient w.r.t. A During backpropogation, gradients at locations which were set to 0 during the forward operation are also zeroed, and otherwise the gradients are scaled up by a factor of $g\frac{1}{1-\alpha}$.

Chapter 3

Activation Operators

3.1 Sigmoid

The Sigmoid function is defined as $\sigma(x) = \frac{1}{1+e^{-x}}$ To find the gradient, we will apply the chain rule and simplify.

$$\frac{d}{dx}\sigma(x) = \frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right] \tag{3.1}$$

$$= \frac{d}{dx}\left(1+\mathrm{e}^{-x}\right)^{-1} \tag{3.2}$$

$$= -(1+e^{-x})^{-2}(-e^{-x}) \tag{3.3}$$

$$= \frac{e^{-x}}{(1+e^{-x})^2} \tag{3.4}$$

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \tag{3.5}$$

$$= \frac{1}{1+e^{-x}} \cdot \frac{(1+e^{-x})-1}{1+e^{-x}} \tag{3.6}$$

$$= \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \tag{3.7}$$

$$= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) \tag{3.8}$$

$$= \sigma(x) \cdot (1-\sigma(x)) \tag{3.9}$$

Therefore, we say that the gradient of the sigmoid function with respect to x is $\sigma(x) \cdot (1-\sigma(x))$.

3.2 Softmax

The Softmax operator "...converts a vector of length K into a probability distribution of K possible outcomes".[Wik23b] This operation is commonly used

in classification, to produce a vector of probabilities.

$$\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{K} e^{z_j}}$$

In the example below, you can see that the softmax normalized the outputs to a range [0, 1] and the sum of these values is 1.

$$\sigma([1,2,3])_i = [0.09, 0.24, 0.67]$$

In the following proof, I will refer to the input as z and the output as w, both with a length of K.

Proposition 1: To define $\frac{\delta w}{\delta z}$ we will first define $\frac{\delta w_i}{\delta z_j}$, for which we will apply the quotient rule, defined below.

$$\frac{\delta}{\delta x}(\frac{u}{v}) = \frac{v\frac{\delta u}{\delta x} - u\frac{\delta v}{\delta x}}{v^2}$$

Now we run into a problem. We want to substitute z_j for x, but we will get different answers for $i \neq j$ and $i = j$. First, lets' define the $i = j$ case. For simplicity, $\Sigma = \sum_{j=0}^{K} e^{z_j}$.

$$\frac{\delta w_i}{\delta z_i}(\sigma)_i = \frac{\Sigma\frac{\delta e^{z_i}}{\delta z_i} - e^{z_i}\frac{\delta \Sigma}{\delta z_i}}{\Sigma^2} = \frac{e^{z_i}\Sigma - e^{z_i}e^{z_i}}{\Sigma^2}$$

Simplify to an easier form.

$$= \frac{e^{z_i}}{\Sigma}\frac{\Sigma - e^{z_i}}{\Sigma} = \sigma_i(1 - \sigma_i)$$

Proposition 2: In proposition 1 we defined $\frac{\delta w_i}{\delta z_j}$ when $i = j$, so now let us define the $i \neq j$ case. We will again apply the quotient rule.

$$\frac{\delta w_i}{\delta z_j}(\sigma) = \frac{0 - e^{z_i}\frac{\delta \Sigma}{\delta z_i}}{\Sigma^2} = \frac{-e^{z_j}e^{z_i}}{\Sigma^2} = -\frac{e^{z_j}}{\Sigma}\frac{e^{z_i}}{\Sigma} = -\sigma_j\sigma_i$$

Therefore, we can define $\frac{\delta w_i}{\delta z_j}$ as follows:

$$\begin{cases} i = j & \sigma_i(1 - \sigma_j) \\ i \neq j & -\sigma_j\sigma_i \end{cases}$$

Chapter 4

Higher Rank Operators

4.1 Matmul

The Matmul operator computes the dot product of two matrices **A** and **B**. When **A** is an M x N matrix, **B** is an N x P matrix, the product **C** is a M x P matrix defined as:

$$\sum_{i=0}^{M}\sum_{j=0}^{P}\boldsymbol{C}_{ij} = \sum_{k=0}^{N}\boldsymbol{A}_{ik}\boldsymbol{B}_{kj}$$

Consider the following row-major matrices, where indices start at 0 and 0,0 is the top left.

$$A = \begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} B = \begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix} C = \begin{bmatrix}22 & 28\\ 49 & 62\end{bmatrix}$$

Proposition 1: The dot product of A and B is denoted $C = A \cdot B$. The definition of an element C_{ij} is the dot product of row vector A_i and column vector B_j.

$$C_{ij} = \sum_{k=0}^{N} A_{ik}B_{kj}$$

For example, to find C_{00}, this formula will expand to:

$$C_{00} = A_{00} * B_{00} + A_{01} * B_{10} + A_{02} * B_{20}$$

Proposition 2: To find the gradient of C w.r.t. an element of A or B, we will first differentiate the formula defined by proposition 1 by applying the product rule.

$$\frac{\delta}{\delta X}(\sum_{k=0}^{N} A_{ik}B_{kj}) = \sum_{k=0}^{N} B_{kj}\frac{\delta A_{ik}}{\delta X} + A_{ik}\frac{\delta B_{kj}}{\delta X}$$

Now substitute to find the gradient of C_{ij} w.r.t. A_{ik} and B_{kj}. We can remove the summation because we are finding the gradient for an element of X instead

of the entire vector X.

$$\frac{\delta}{\delta A_{ik}}(A_{ik}B_{kj}) = B_{kj}\frac{\delta A_{ik}}{\delta A_{ik}} + A_{ik}\frac{\delta B_{kj}}{\delta A_{ik}} = B_{kj}$$

$$\frac{\delta}{\delta B_{kj}}(A_{ik}B_{kj}) = B_{kj}\frac{\delta A_{ik}}{\delta B_{kj}} + A_{ik}\frac{\delta B_{kj}}{\delta B_{kj}} = A_{ik}$$

In plain english, the gradient of an index with respect to some C_{ij} is the element in the corresponding vector it was multiplied by to get C_{ij}. For example, the gradient of A_{01} w.r.t. C_{00} is B_{10}, since A_{01} is multiplied by B_{10} while calculating C_{00}. Also, A_{01} was also multiplied by B_{11} while calculating C_{01}. Since we know all the elements A_{01} was multiplied by, we can define the gradient of C w.r.t. A_{01} to be $B_{10} + B_{11}$.

Take note of the fact that the gradient w.r.t. A_{01} is the sum of the elements of row vector B_1. The same logic applies for e.g. the gradient w.r.t. B_{20}, which is the sum of the elements in column vector A_2. Therefore, we can define the gradient of C w.r.t. A_{ik} as the sum of the columns of B_k and the gradient of C w.r.t. A_{kj} as the sum of the rows of A_k.

$$\frac{\delta}{\delta A_{ik}} = \sum_{j=0}^{P} B_{kj}$$

$$\frac{\delta}{\delta B_{kj}} = \sum_{i=0}^{M} A_{ik}$$

Proposition 3: To calculate the gradient w.r.t. A we can take advantage of the axiom that A shares an axis N with B and axis M with C. We can then take the dot product of some gradient matrix GC and the transpose of B. The same logic applies to the gradient w.r.t. B, taking advantage of the axiom that B shares an axis N with A and axis P with C. Therefore, we can define the gradients as follows:

$$GA = GC \cdot B^T$$

$$GB = A^T \cdot GC$$

Here are those operations written out:

$$GA = \begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} \cdot \begin{bmatrix}1 & 3 & 5\\ 2 & 4 & 6\end{bmatrix} = \begin{bmatrix}3 & 7 & 11\\ 3 & 7 & 11\end{bmatrix}$$

$$GB = \begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix} \cdot \begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} = \begin{bmatrix}5 & 5\\ 7 & 7\\ 9 & 9\end{bmatrix}$$

This satisfies proposition 2, since GA is the sum of the rows of B, and GB is the sum of the rows in A.

GC here is the input gradient w.r.t. C. While backpropogating, these may be any value, but for this example case it is an identity matrix. GC is assumed to have the same shape as C. The output gradient GA has the same dimensions as A, and GB has the same dimensions as B.

Chapter 5

Loss Functions

5.1 MSE

The Mean Squared Error function measures the average of the squares of the losses. It is defined as follows, where t and p are vectors of length n. t is the target output and p is the predicted output.

$$mse(t, p) = \frac{1}{n}\sum_{i=0}^{n}(t_i - p_i)^2$$

Gradient The MSE of an estimator is defined as below[Wik23a].

$$mse(\hat{\theta}) = E_\theta[(\hat{\theta} - \theta)^2]$$

To find the gradient w.r.t. $\hat{\theta}$, we can apply the chain rule.

$$\frac{\delta}{\delta\hat{\theta}}((\hat{\theta} - \theta)^2) = 2(\hat{\theta} - \theta)$$

Then we can rewrite this in terms of t and p.

$$\frac{\delta}{\delta\hat{p_i}}((\hat{p_i} - t_i)^2) = 2(\hat{p_i} - t_i)$$

Therefore, the gradient of the MSE of an estimator is defined as $2(\hat{p_i} - t_i)$.

5.2 MAE

The Mean Absolute Error function measures the average of the absolute value of the losses. It is defined as follows, where t and p are vectors of length n. t is the target output and p is the predicted output.

$$mae(t, p) = \frac{1}{n}\sum_{i=0}^{n}|p_i - t_i|$$

Gradient The MAE of an estimator is defined below.

$$mae(\hat{\theta}) = E_\theta[|\theta - \hat{\theta}|]$$

Bibliography

[Wik23a] Wikipedia. *Mean squared error — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Mean%20squared%20error&oldid=1179991528. [Online; accessed 17-October-2023]. 2023.

[Wik23b] Wikipedia. *Softmax function — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Softmax%20function&oldid=1178481093. [Online; accessed 12-October-2023]. 2023.