Hardmax

Softmax vs Argmax vs Hardmax
Softmax

```
>>> input_array = mx.nd.array([[3., 0.5, -0.5, 2., 7.],
>>>                             [2., -.4, 7.,   3., 0.2]])
>>> softmax_act = mx.nd.SoftmaxActivation(input_array)
>>> print softmax_act.asnumpy()
[[  1.78322066e-02   1.46375655e-03   5.38485940e-04   6.56010211e-03   9.73605454e-01
 [  6.56221947e-03   5.95310994e-04   9.73919690e-01   1.78379621e-02   1.08472735e-03
```

Argmax - Returns indices of the maximum values along an axis

```
x = [[ 0.,  1.,  2.],[ 3.,  4.,  5.]]
// argmax along axis 0argmax(x, axis=0) = [ 1.,  1.,  1.]
// argmax along axis 1argmax(x, axis=1) = [ 2.,  2.]
// argmax along axis 1 keeping same dims as an input array
// argmax(x, axis=1, keepdims=True) = [[ 2.],[ 2.]]
```

Hardmax -

```
>>> xn
[[[[2. 3. 4.]
[1. 2. 3.]]
[[1. 1. 1.]
[1. 2. 3.]]]

[[[4. 5. 6.]
[4. 4. 4.]]
[[1. 1. 1.]
[1. 2. 3.]]]]
<NDArray 2x2x2x3 @cpu(0)>
>>> mx.nd.contrib.hardmax(xn)

[[[[0. 0. 1.]
[0. 0. 1.]]
[[1. 0. 0.]
[0. 0. 1.]]]

[[[0. 0. 1.]
[1. 0. 0.]]
[[1. 0. 0.]
[0. 0. 1.]]]]
<NDArray 2x2x2x3 @cpu(0)>
```

Front-end implementation of hardmax

- Ndarray
- Symbol

Back-end implementation of hardmax

Model that uses hardmax?

1. https://cntk.ai/pythondocs/CNTK_204_Sequence_To_Sequence.html?highlight=identity

Tensorflow repo -
hardmax
2 files - https://github.com/tensorflow/tensorflow/search?q=hardmax&unscoped_q=hardmax

CNTK
Hardmax, Softmax - Activation functions
The `Hardmax()` operation determines the element with the highest value and represents its location as a one-hot vector/tensor. This is used for performing classification.
https://github.com/anwald/CNTK/wiki/Times-and-TransposeTimes

ONNX
https://github.com/onnx/onnx/blob/master/docs/Operators.md#hardmax
Fix the doc -

Paddle
https://github.com/PaddlePaddle/Paddle/pull/9228/files

1. Custom OP
2. MLP
3. Complicated model

hardmax, hardmax nlp, rl - poor search
hard attention

max, and therefore ReLU, maxout and max pooling, are continuous and almost everywhere differentiable. This is enough to use them with gradient descent optimization.
Argmax is not continuous and can't be used with standard gradient descent techniques. If you want to use it in neural networks (e.g. in "hard" attention models) you typically have to use some kind of Monte Carlo optimization algorithm, such as REINFORCE. Otherwise you can replace argmax with softmax, which is continuous and differentiable, as typically done in "soft" attention models.

Randint
Numpy needs 1 value (low is not optional, rest all params are) - https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.randint.html