SciSharp

A competitive ML Ecosystem for .NET

By Meinrad Recheis

If this is Machine Learning in Python ...

… then this is Machine Learning in C#

SciSharp 

Our mission is to to change that.

- SciSharp is an Open Source Organization
- An international team of skilled C# developers with ML background
- The goal: make major ML frameworks available to .NET community

Stack Overflow Developer Survey SciSharp

„Most Loved Other Frameworks, Libraries and Tools“

Makes sense to bring loved developer tools closer together

SciSharp

TensorFlow.NET

TensorFlow.NET

SciSharp #

A C# port of TF with bindings to the highly efficient C++ CPU/GPU computation layers

Status of TensorFlow.NET

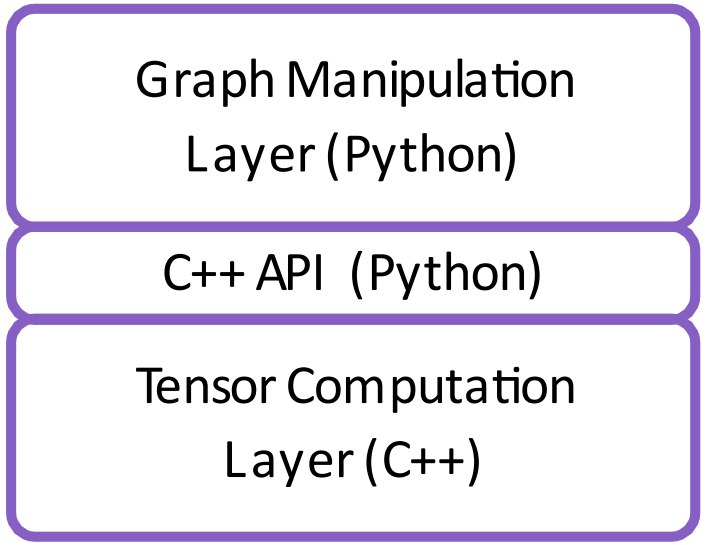
- Inference [stable, ready to use]
- Training [beta, works for many models]
- Building models from scratch in C# [alpha, works for some models]
- 106,327 lines of code
- 3,412 downloads on Nuget as of June 2019

TensorFlow.NET

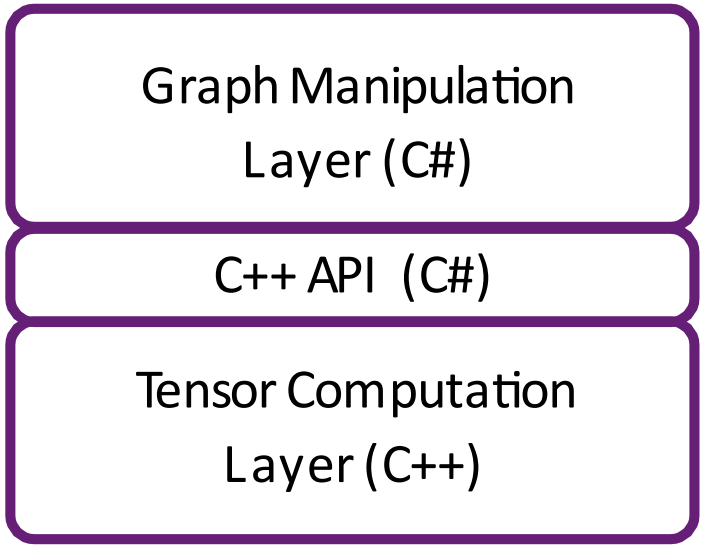
SciSharp

Architecture and Comparison

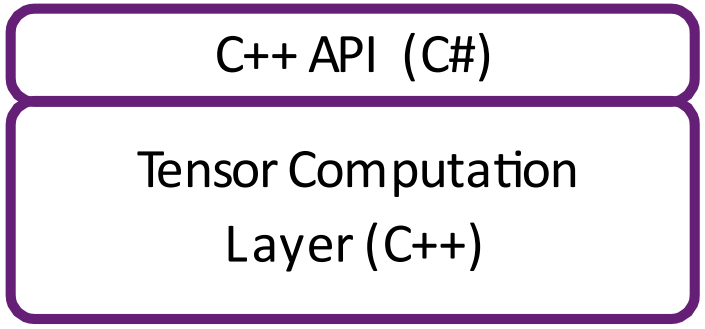
TensorFlow



TensorFlow.NET



TensorFlowSharp
(by Microsoft)

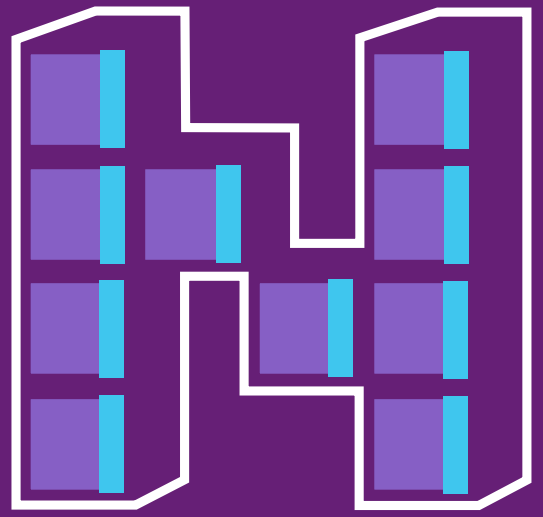


TensorFlow.NET

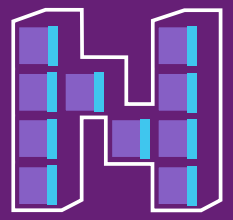
SciSharp

```
40      var b = tf.Variable(-0.73f, name: "bias");
41
42      // Construct a linear model
43      var pred = tf.add(tf.multiply(X, W), b);
44
45      // Mean squared error
46      var cost = tf.reduce_sum(tf.pow(pred - Y, 2.0f)) / (2.0f * n_samples);
47
48      // Gradient descent
49      // Note, minimize() knows to modify W and b because Variable objects are traina
50      var optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)
51
52      // Initialize the variables (i.e. assign their default value)
53      var init = tf.global_variables_initializer();
54
55      // Start training
56      with<Session>(tf.Session(), sess =>
```

SciSharp



NumSharp



NumSharp

SciSharp 

A pure C# implementation of NumPy‘s API with interchangable numerical computation engines

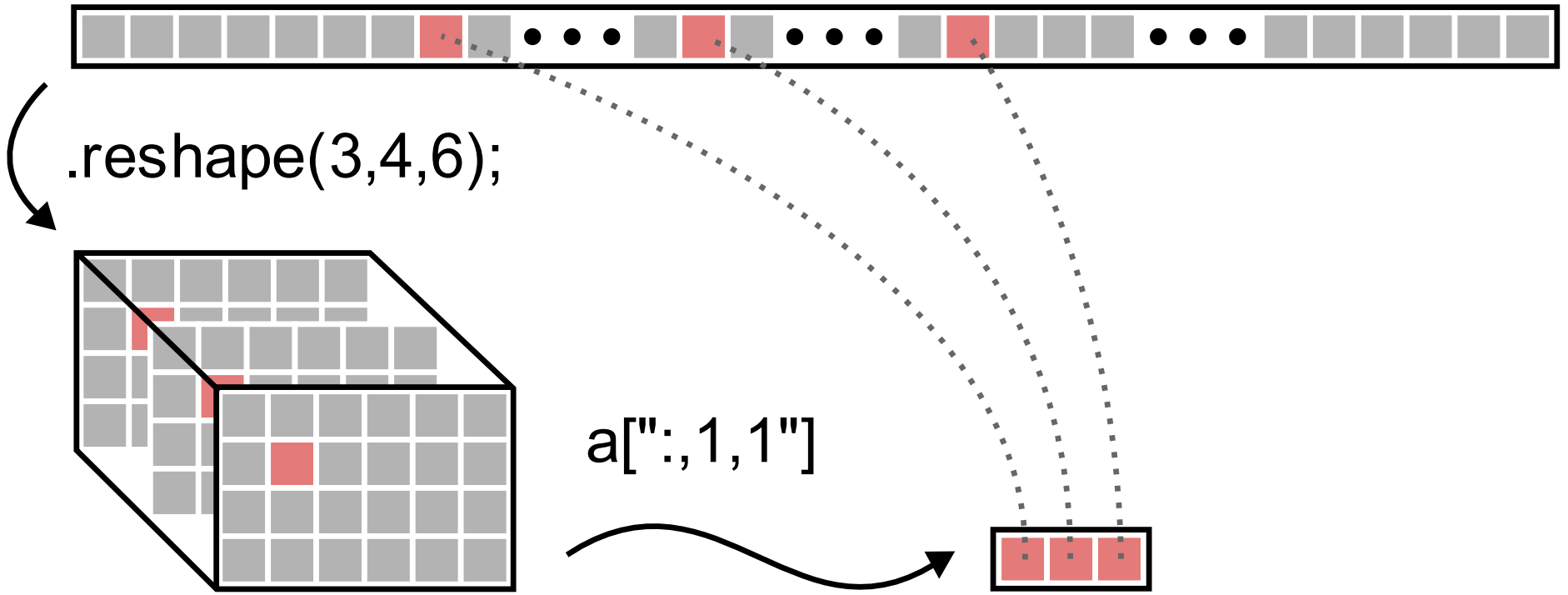
Status of NumSharp [stable, but somewhat incomplete]

- Used by TensorFlow.NET (i.e. for batching)
- Implements a large subset of the NDArray API
- 23,117 lines of code
- 6,992 downloads on Nuget as of June 2019

NumSharp

SciSharp

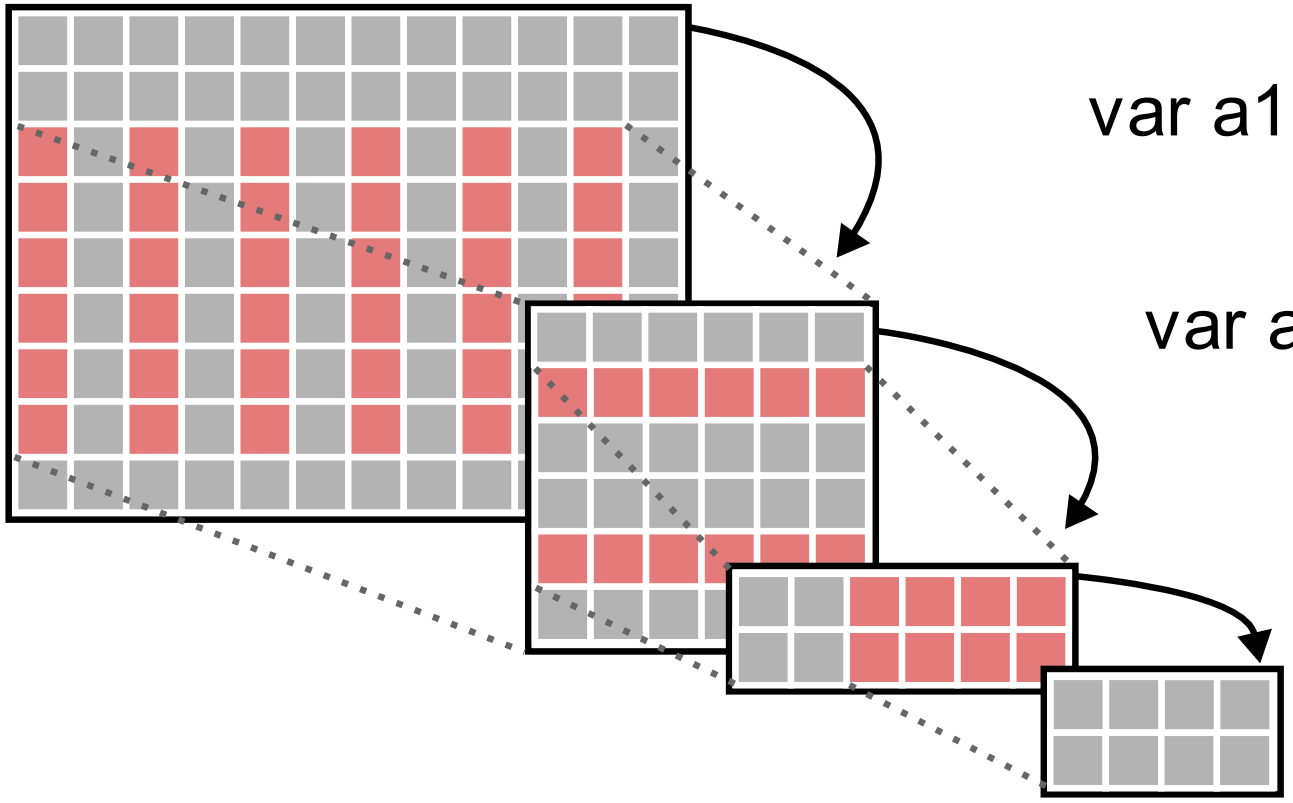
```
var a=np.array(new  double[72])
```



NumSharp

SciSharp

```
var a=np.array(data).reshape(9,12)
```

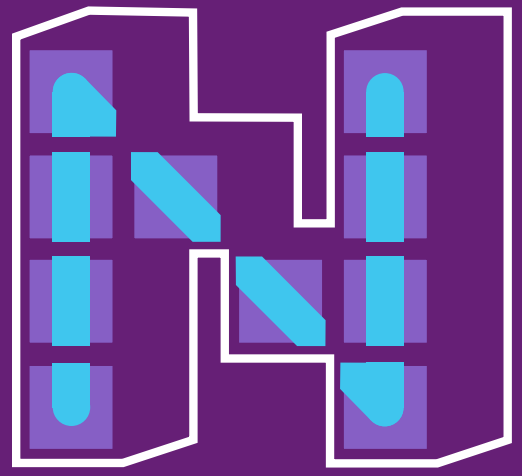


```
var a1=a["2:8,::2"]
```

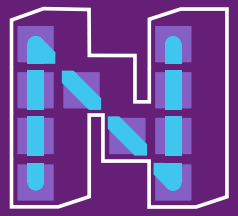
```
var a2=a1["1::3"]
```

```
a2[":,2:"]
```

SciSharp



Numpy.NET

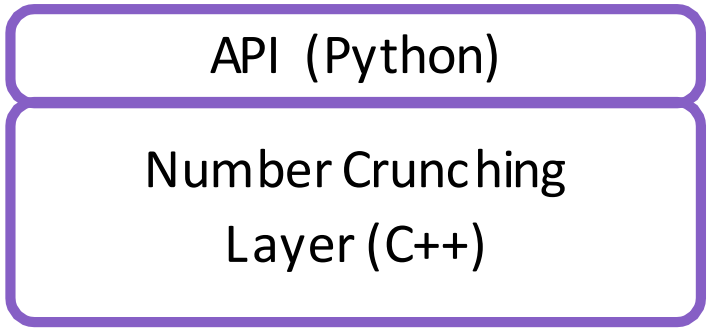


Numpy.NET

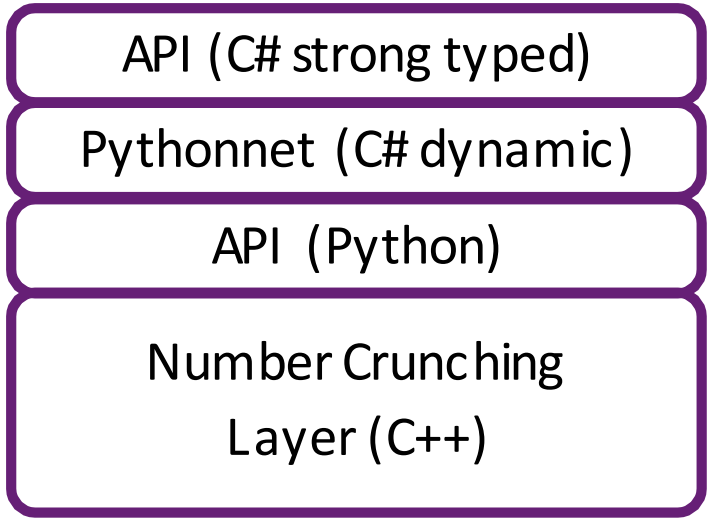
SciSharp

Architecture and Comparison

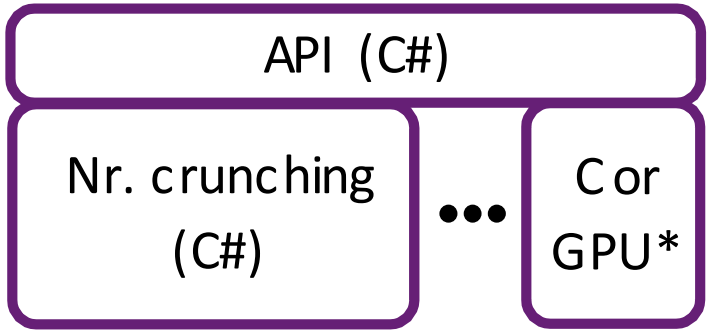
Numpy



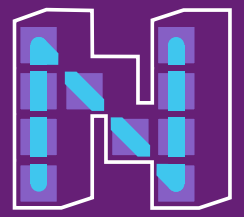
Numpy.NET



NumSharp



*GPU support not implemented yet



Numpy.NET

SciSharp

The most complete .NET bindings for NumPy.

Status of Numpy.NET [stable, ready to use]

- Used by Keras.NET and Torch.NET
- Auto-generated: implements the **complete** NDarray API
- > 500 API functions
- 144,016 lines of code
- Available as a standalone Nuget as well as one that depends on Python

SciSharp

Keras.NET

Keras.NET

SciSharp 

.NET bindings for the high-level NN API running on top of TensorFlow, CNTK, or Theano.

Status of Keras.NET [stable, ready to use]

- Mostly auto-generated: implements ~90% of the Keras API
- 7,573 lines of code
- Several working examples published

Keras.NET

SciSharp 

```
// Build CNN model
var model = new Sequential();
model.Add(new Conv2D(32, kernel_size: (3, 3).ToTuple(), activation: "relu", input_shape: input_shape));
model.Add(new Conv2D(64, (3, 3).ToTuple(), activation: "relu"));
model.Add(new MaxPooling2D(pool_size: (2, 2).ToTuple()));
model.Add(new Dropout(0.25));
model.Add(new Flatten());
model.Add(new Dense(128, activation: "relu"));
model.Add(new Dropout(0.5));
model.Add(new Dense(num_classes, activation: "softmax"));
model.Compile(loss: "categorical_crossentropy", optimizer: new Adadelta(), metrics: new string[] { "accuracy" });
model.Fit(x_train, y_train, batch_size: batch_size, epochs: epochs, verbose: 1,
          validation_data: new NDarray[] { x_test, y_test });          //   ← Numpy.NET
var score = model.Evaluate(x_test, y_test, verbose: 0);
```

More libraries coming soon

SciSharp

- Torch.NET — 10k lines of the torch.Tensor API already generated
- SpaCy.NET — Natural Language Processing
- Pandas.NET — Time series and structured data analysis
- SciSharpCube — Docker cube with pre-installed SciSharp tools
- ICSharpCore — C# plugin for Jupyter Notebook
- SiaNet — An easy to use CUDA enabled Deep Learning Library
- scikit-learn.net — C# implementation of scikit-learn
- Ludwig.NET — Very simple Machine Learning
- Gym.NET — Reinforcement Learning Environments

Reasons to join us

SciSharp

- Gain deep understanding of a variety of ML/AI frameworks
- Make a name for yourself in the Open Source community
- Karma ;)

If you want to reach out to us join our chat on Gitter

https://gitter.im/sci-sharp/community

Thank you

SciSharp 

https://github.com/SciSharp