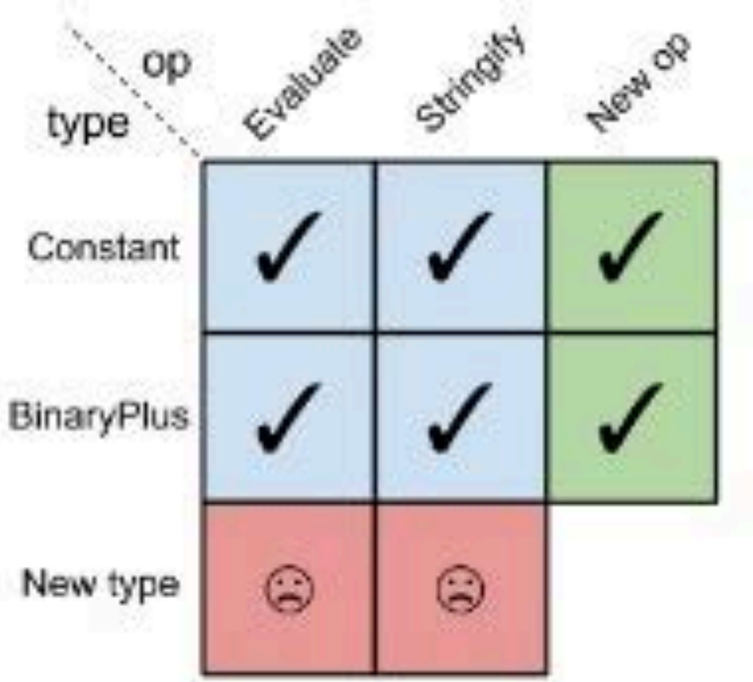

ThoughtWorks®
THE EXPRESSION PROBLEM
automated learning?

THE EXPRESSION PROBLEM



Attila Magyar, 2016
Microsec ltd.

HYPERPARAMETER IN DEEPLEARNING.SCALA

```
val hyperparameters = Factory[
  INDArrayWeights with INDArrayLayers with FixedLearningRate
].newInstance(
  learningRate = 0.0001
)

val w: hyperparameters.INDArrayWeight = hyperparameters.INDArrayWeight(Nd4j.randn(3072, 64))

def myLayer(input: hyperparameters.INDArrayLayer): hyperparameters.INDArrayLayer = {
  input dot w
}
```

HYPERPARAMETER IN DEEPLEARNING.SCALA

```
val hyperparameters = Factory[
  INDArrayWeights with INDArrayLayers with FixedLearningRate
].newInstance(
  learningRate = 0.0001
)
import hyperparamters.{INDArrayWeight, INDArrayLayer}


val w: INDArrayWeight = INDArrayWeight(Nd4j.randn(3072, 64))
def myLayer(input: INDArrayLayer): INDArrayLayer = {
  input dot w
}
```

MULTIPLE CONFIGURATIONS OF HYPERPARAMETERS

```
val hyperparameters1 = Factory[Builtins with FixLearningRate].newInstance(
  learningRate = 0.001
)
val weight1 = hyperparameters1.INDArrayWeight(Nd4j.randn(10, 10))
val layer1(input: hyperparameters1.INDArrayLayer) = {
  tanh(input dot weight1)
}
```

```
val hyperparameters2 = Factory[Builtins with FixLearningRate].newInstance(
  learningRate = 0.002
)
val weight2 = hyperparameters2.INDArrayWeight(Nd4j.randn(10, 10))
val layer2(input: hyperparameters1.INDArrayLayer) = {
  tanh(layer1(input) dot weight2)
}
```

MULTIPLE CONFIGURATIONS OF HYPERPARAMETERS

```
val hyperparameters1 = Factory[Builtins with FixLearningRate].newInstance(
  learningRate = 0.001
)
val weight1 = hyperparameters1.INDArrayWeight(Nd4j.randn(10, 10))
val layer1[Input](input: Input): hyperparameters1.INDArrayLayer = {
  tanh(input dot weight1)
}

val hyperparameters2 = Factory[Builtins with FixLearningRate].newInstance(
  learningRate = 0.002
)
val weight2 = hyperparameters2.INDArrayWeight(Nd4j.randn(10, 10))
val layer2[Input](input: Input): hyperparameters2.INDArrayLayer = {
  tanh(input dot weight2)
}
```

ENABLING A PLUGIN FOR NEW FUNCTIONS

```
val hyperparameters = Factory[Builtins with CNNs].newInstance()
import hyperparameters.{INDArrayWeight, INDArrayLayer, conv2d}

val weight: INDArrayWeight = INDArrayWeight(Nd4j.randn(3, 64, 3, 3))
val bias: INDArrayWeight = INDArrayWeight(Nd4j.zeros(64))
def myLayer1(input: INDArrayLayer): INDArrayLayer = {

  conv2d(input, weight, bias, (3, 3), (1, 1), (1, 1))
}
```

ENABLING A PLUGIN CHANGING EXISTING BEHAVIOR

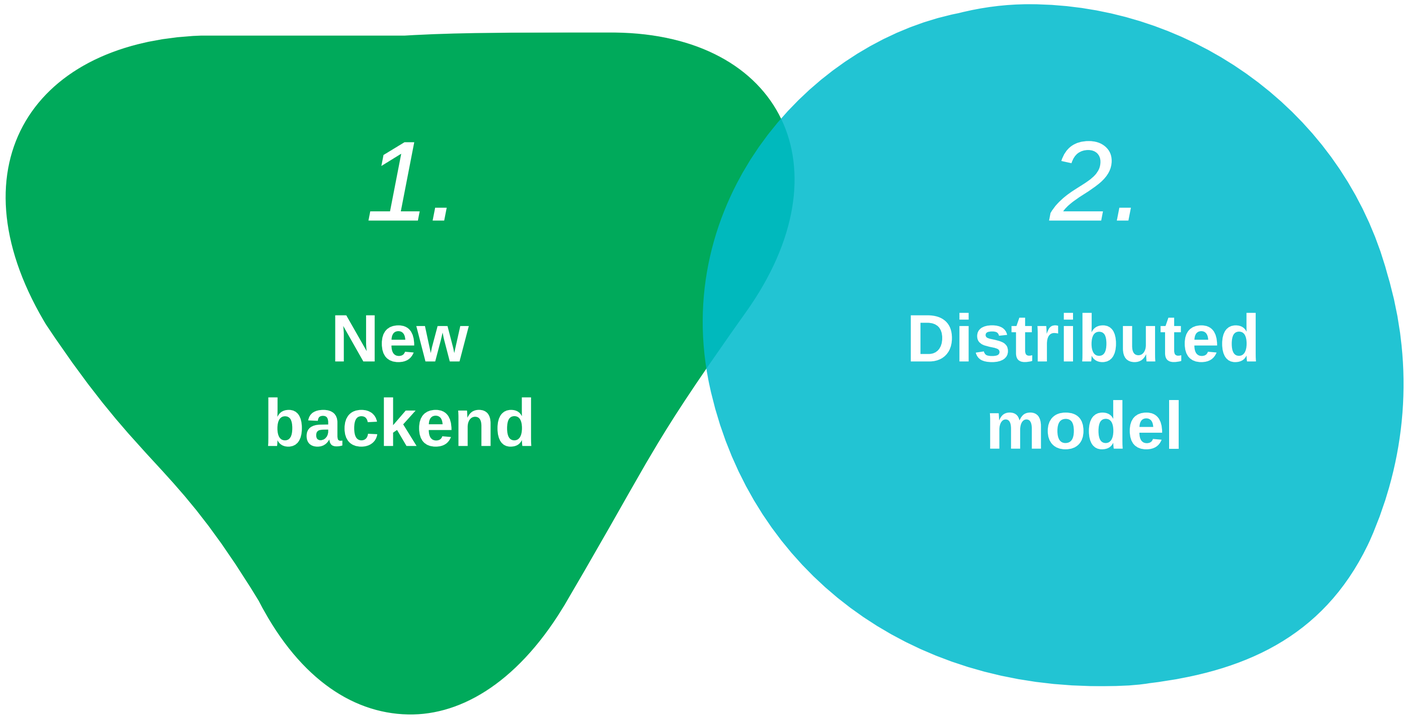
```
val hyperparameters = Factory[
  Builtins with FixedLearningRate with Momentum
].newInstance(
  learningRate = 0.0001,
  mu = 0.97

)
import hyperparamters.{INDArrayWeight, INDArrayLayer}

val w: INDArrayWeight = INDArrayWeight(Nd4j.randn(3072, 64))
def myLayer(input: INDArrayLayer): INDArrayLayer = {
  input dot w
}
```

ThoughtWorks®

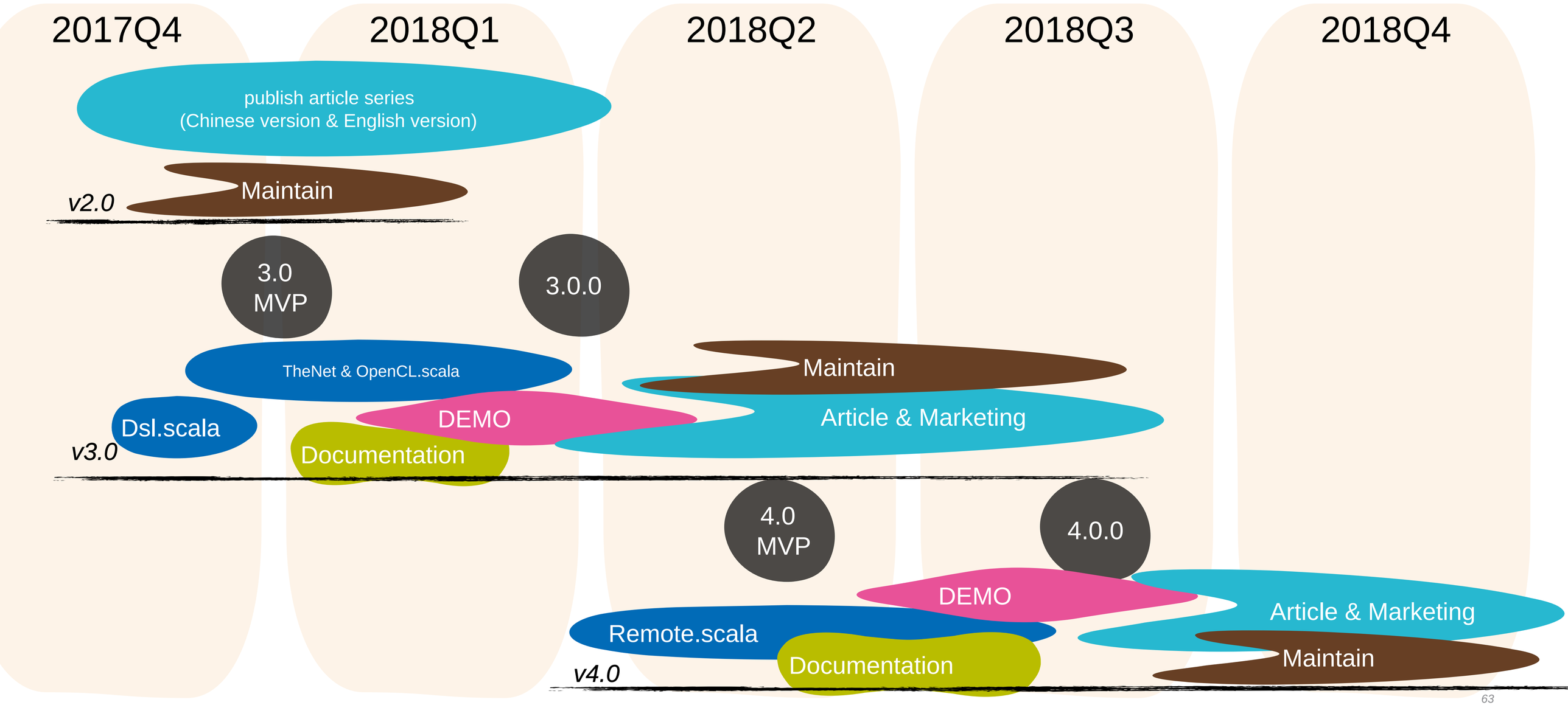
FUTURE WORK

Tomorrow

FUTURE WORK IN DEEPLEARNING.SCALA 3.0



FUTURE WORK


2017Q4
2018Q1
2018Q2
2018Q3
2018Q4
publish article series
(Chinese version & English version)
Maintain
v2.0
3.0
MVP
3.0.0
TheNet & OpenCL.scala
Dsl.scala
DEMO
Documentation
Maintain
Article & Marketing
v3.0
4.0
MVP
4.0.0
DEMO
Remote.scala
Documentation
Article & Marketing
Maintain
v4.0

Q & A

ThoughtWorks®

4500+ technologists with 42 offices in 15 countries

atryyang@thoughtworks.com
xlewang@thoughtworks.com

https://www.thoughtworks.com/deeplearning-scala

人工智能领域工程能力最强的团队

CIES@ThoughtWorks