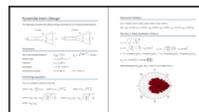




Introduction

Maplesoft éditeur de logiciel mathématiques depuis 30 ans
Partenaire historique des établissements et des IUT.
Modèles de licences flexibles et adaptables.



Maple

Logiciel mathématiques puissant qui combine des algorithmes et outils de traitement de texte pour créer des documents techniques interactifs.

Manipulations symboliques et numériques

$$\sum_{n=0}^m \left(\int_1^n x \, dx \right)$$

Visualisation des problèmes:

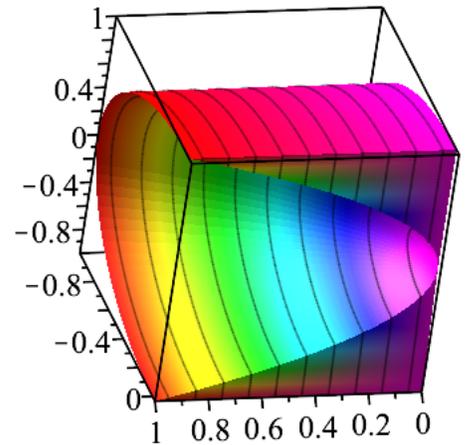
Volume donné par

$$V = \pi \int_0^1 (R^2(y) - r^2(y)) \, dy$$

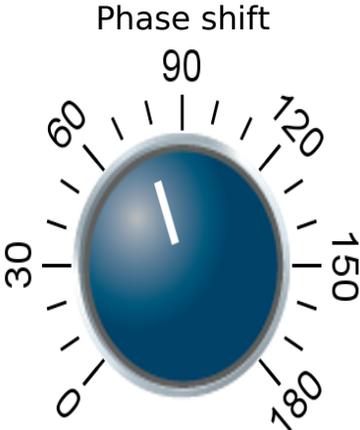
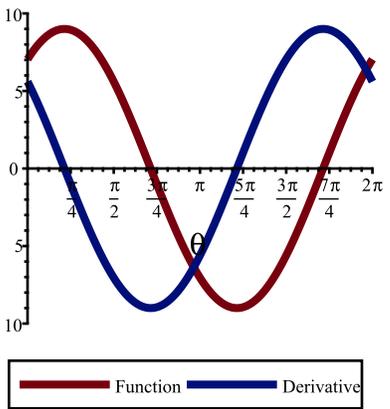
Dans ce problème, $R=1$ et $r=x(y)=\sqrt{y}$. Nous

substituons ces valeurs dans l'expression précédente:

$$V = \pi \int_0^1 (1 - y) dy = \frac{\pi}{2}$$



Applications interactives

<p>Phase shift</p>  <p>Amplitude</p> 	<p>Fonction</p> $9 \sin\left(\theta + \frac{103 \pi}{360}\right)$ <p>Derivative</p> $9 \cos\left(\theta + \frac{103 \pi}{360}\right)$	
--	---	--

Maple pour l'enseignement

with(Student-Calculus1) :

$$6 + x - x^2$$

Applications Machine Learning

[DeepLearning pour la prédiction des feux de forêt](#)

Conclusion

Présentation de la gamme de produits Maplesoft demain Vendredi 11 Juin à 14h



Prediction des feux de forêt grâce au Machine Learning

Sophie Tan
Maplesoft



Introduction

Les données et observations des feux de forêt du Nord du Portugal proviennent de ce document: [UCI Machine Learning Repository](#) ainsi que la description des données utilisées.

- Nous n'utiliserons pas les colonnes mois et jours dans cette application.
- Les 10 premières colonnes sont des indices numériques continus qui caractérisent chaque feu de forêt (endroit, données météorologiques..). La dernière colonne correspond à la surface qui a été brûlée.

Package Deep Learning

Dans cette application nous allons utiliser le package [DeepLearning](#) afin de:

- Former un réseau neuronal avec les données des feux de forêt.
- Prédire la surface brûlée pour un endroit donné en fonction des conditions environnementales et écologiques.

Ce module utilise la plateforme Google TensorFlow. Logiciel open source pour machine learning développé par Google.

Le package DeepLearning est une interface pour donner accès à Google TensorFlow dans Maple grâce à l'API Python de TensorFlow.

Import de données

```
> restart;
> train_data := Import("this:///forestfires_train.csv");
```

	<i>X</i>	<i>Y</i>	<i>FFMC</i>	<i>DMC</i>	<i>DC</i>	<i>ISI</i>	<i>temp</i>	<i>RH</i>	...	
<i>train_data</i> :=	1	7	5	86.2	26.2	94.3	5.1	8.2	51	...
	2	7	4	90.6	35.4	669.1	6.7	18	33	...
	3	7	4	90.6	43.7	686.9	6.7	14.6	33	...
	4	8	6	91.7	33.3	77.5	9	8.3	97	...
	5	8	6	89.3	51.3	102.2	9.6	11.4	99	...
	6	8	6	92.3	85.3	488	14.7	22.2	29	...
	7	8	6	92.3	88.9	495.6	8.5	24.1	27	...
	8	8	6	91.5	145.4	608.2	10.7	8	86	...

(3.1)

```
> test_data := Import("this:///forestfires_test.csv");
```

	<i>X</i>	<i>Y</i>	<i>FFMC</i>	<i>DMC</i>	<i>DC</i>	<i>ISI</i>	<i>temp</i>	<i>RH</i>	...	
<i>test_data</i> :=	1	7	5	93.7	231.1	715.1	8.4	26.4	33	...
	2	5	4	93.6	235.1	723.1	10.1	24.1	50	...
	3	8	6	94.8	222.4	698.6	13.9	27.5	27	...
	4	6	3	92.7	164.1	575.8	8.9	26.3	39	...
	5	6	5	93.4	17.3	28.3	9.9	13.8	24	...
	6	2	4	92	203.2	664.5	8.1	24.9	42	...
	7	2	5	91.6	181.3	613	7.6	24.8	36	...
	8	8	8	91.7	191.4	635.9	7.8	26.2	36	...

(3.2)

```
> cols := ColumnLabels(train_data)[-2];
      cols := [X, Y, FFMC, DMC, DC, ISI, temp, RH, wind, rain]
```

(3.3)

Nous avons ici 517 entrées dont 413 utilisées pour former le réseau neuronal et 104 pour le tester.

Former le réseau neuronal et évaluation du DNR

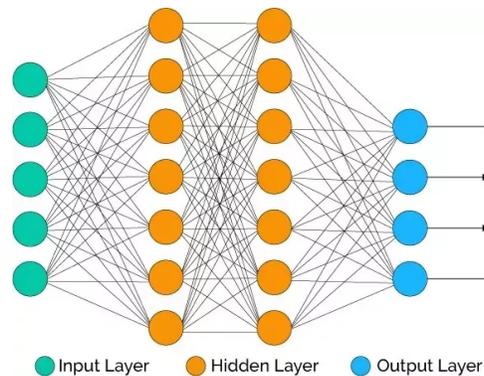
Chargement du package DeepLearning

```

> with(DeepLearning) :
> fc := [seq(NumericColumn(convert(u, string), shape = [1]), u in
  cols)];
fc := [[[Feature Column],
  [NumericColumn(key='cols', shape=(1), default_value=None, dtype=tf.float32,
  normalizer_fn=None)]]]
  
```

(4.1)

Dans ce modèle nous définissons 3 couches de neurones cachées avec 32, 64, et 32 noeuds.



```

> regressor := DNNRegressor(fc, hidden_units=[32,64,32]):
> regressor:-Train(train_data[1..10], train_data[11], steps = 20,
  num_epochs = 20, shuffle = false):
  
```

Nous spécifions que les entrées vont être passées 20 fois dans le modèle et chaque itérations consiste en 20 étapes.

Une fois que nous avons formé le réseau neuronal il est nécessaire de l'évaluer avec des données test. Cela permet de vérifier la précision du réseau neuronal et d'ajuster le modèle si nécessaire.

```

> regressor:-Evaluate(test_data[1..10], test_data[11], steps = 200)
;
  
```

"average_loss"	6448.77685546875	(4.2)
"loss"	825443.437500000	
"global_step"	60	

Construction de la fonction prédictive

```

> predictor := proc (ds) regressor:-Predict(Transpose(DataFrame(ds)), num_epochs = 1, shuffle =
  false)[1] end proc;
predictor := proc(ds) (5.1)
  regressor:-Predict(Transpose(DataFrame(ds)), num_epochs = 1, shuffle = false) [ 1 ]
end proc

```

Ci-dessus nous avons construit une fonction prédictive qui prend en compte des données et retourne une prédiction basée sur le réseau neuronal formé.

Et maintenant nous pouvons passer des informations d'un feu de forêt à un endroit donné avec des conditions météorologiques spécifiques pour prédire la surface brûlée par ce feu.

```

> ds := DataSeries([4,4,88.1,25.7,67.6,3.8,14.1,43,2.7,0], labels=
  cols);
> predictor(ds);
(5.2)
  [ "predictions" [ 3.95885300636292 ] ]

```

Les prédictions suggèrent que ce feu de forêt dans les conditions données brûlera une surface de ~3.96 ha d'après notre modèle.