

Введение в нейронные сети.

Урок 3. TensorFlow



GeekBrains

План вебинара



1. Инструменты для создания нейронных сетей.
2. Общие сведения о TensorFlow
3. Синтаксис TensorFlow
4. Практика





Место TensorFlow среди др. инструментов

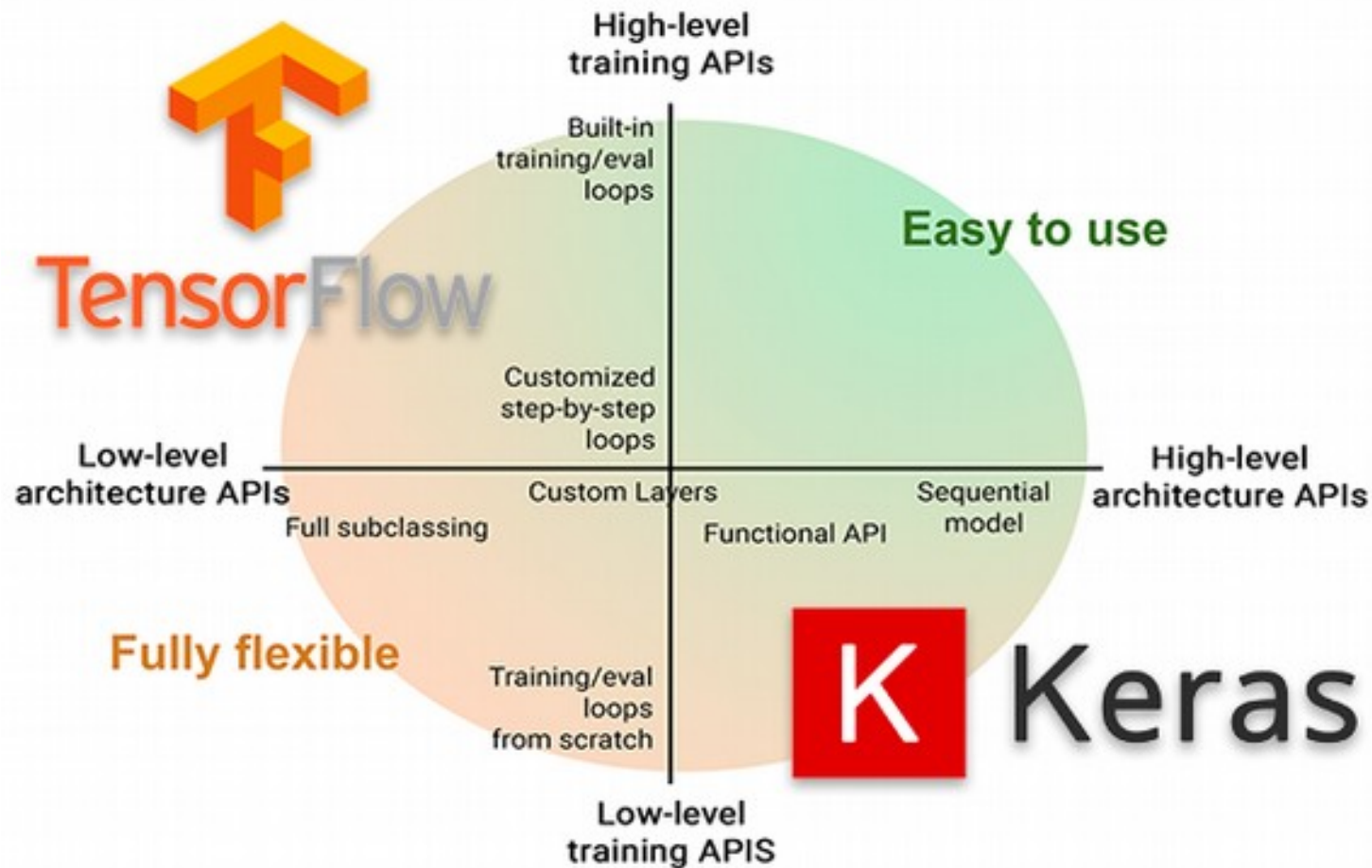
GeekBrains



Сравнение с Keras



GeekBrains



TensorFlow 2.0



Easy

Simplified APIs.
Focused on Keras and
eager execution



Powerful

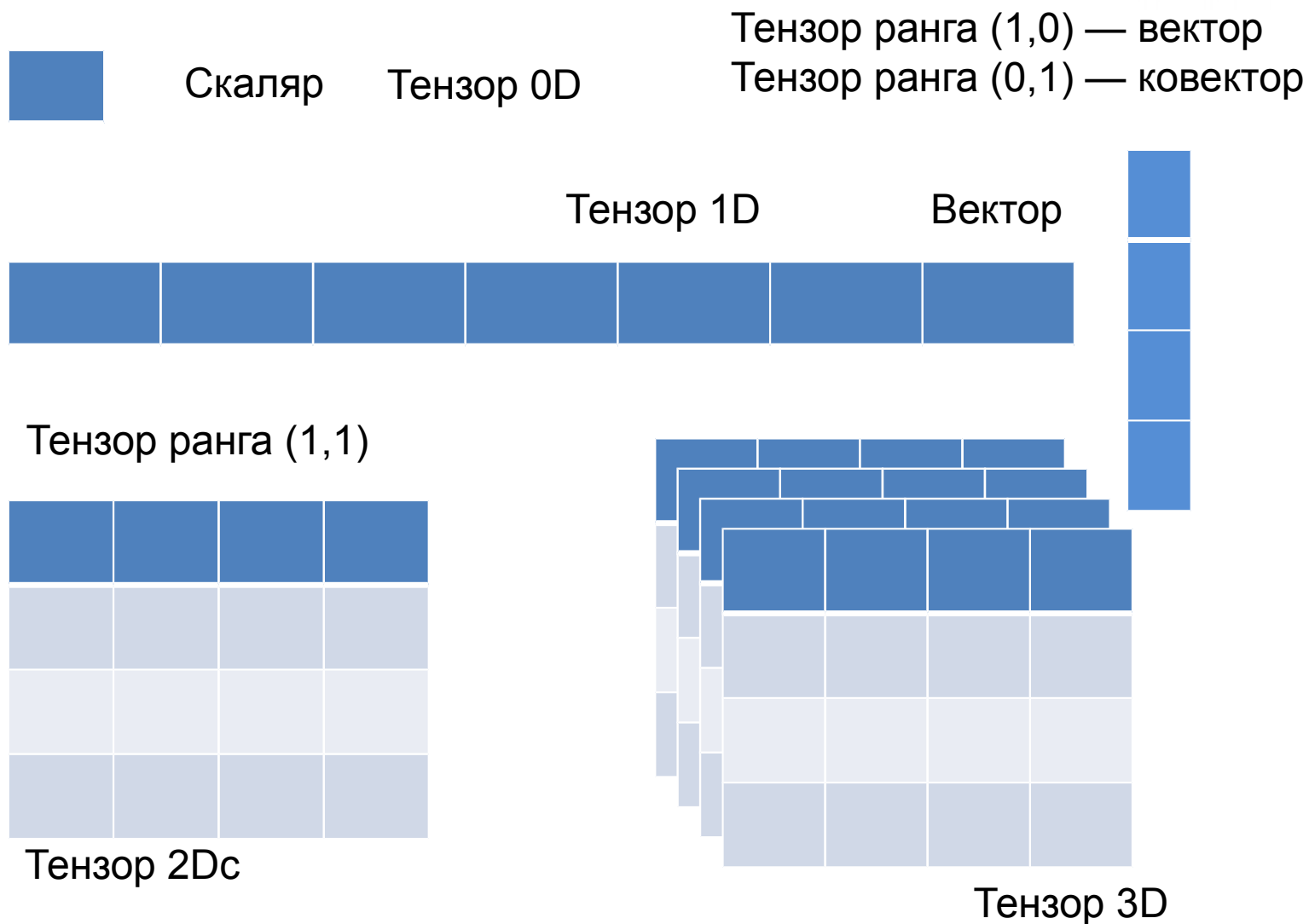
Flexibility and performance.
Power to do cutting edge research
and scale to > 1 exaflops



Scalable

Tested at Google-scale.
Deploy everywhere

Основы синтаксиса - Тензор





Основы синтаксиса - Тензор

- Сумма тензоров
- Сумма двух векторов это тензор 1-го ранга

$$A = \langle a_1, a_2, a_3 \rangle, B = \langle b_1, b_2, b_3 \rangle$$

- Тензор 1 ранга

$$C = A + B = \begin{bmatrix} a_1 + b_1 & a_2 + b_2 & a_3 + b_3 \end{bmatrix}$$



Основы синтаксиса - Тензор

- Произведение тензора и скаляра
- Произведение вектора (тензор 1-го ранга) и B

$$A = \langle a_1, a_2, a_3 \rangle, B$$

- Тензор 1-го ранга

$$C = A B = \begin{bmatrix} Ba_1 & Ba_2 & Ba_3 \end{bmatrix}$$



Основы синтаксиса - Тензор

- Произведение тензоров
- Произведение двух векторов это тензор 2-го ранга

$$A = \langle a_1, a_2, a_3 \rangle, B = \langle b_1, b_2, b_3 \rangle$$

- Тензор второго ранга - *диада*

$$C = A^T B = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}$$



Основы синтаксиса - Тензор

- Операции над тензорами приводят к тензорам
- Произведение двух тензоров ранга n и ранга m - тензор $n+m$ -го ранга

$$C_{ksmij} = B_{ksm} * A^{ij}$$

TensorFlow

- У тензоров есть имена.
- Существует понятие формы тензора.
- Тензоры типизированы и типы для них задаются из библиотеки.

Вопросы



GeekBrains



Оптимизаторы

SGD

$$x_{t+1} = x_t - \alpha \nabla f(x_t)$$

```
while True:
    dx = compute_gradient(x)
    x += learning_rate * dx
```

SGD+Momentum

$$v_{t+1} = \rho v_t + \nabla f(x_t)$$

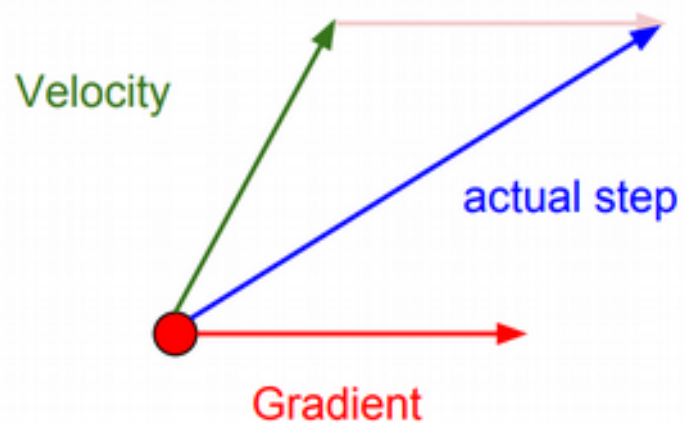
$$x_{t+1} = x_t - \alpha v_{t+1}$$

```
vx = 0
while True:
    dx = compute_gradient(x)
    vx = rho * vx + dx
    x += learning_rate * vx
```

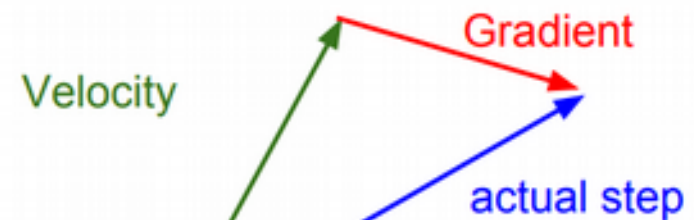
<https://www.reg.ru/blog/stehnfordskij-kurs-lekciya-4-vvedenie-v-nejronnye-seti/>

Оптимизаторы

Momentum update:



Nesterov Momentum

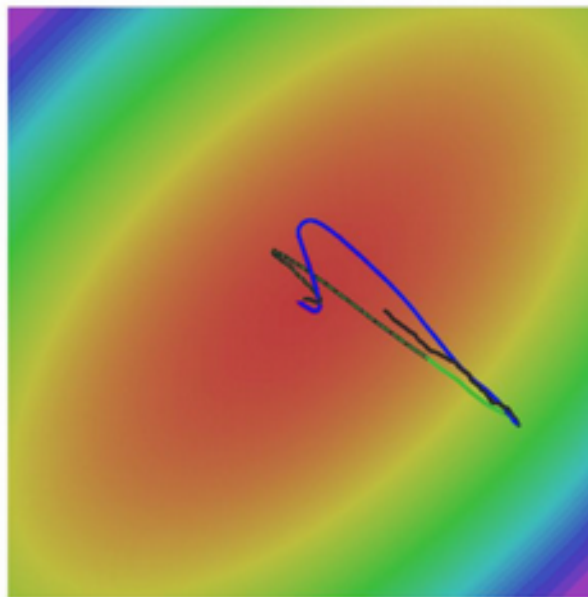


Оптимизаторы

Nesterov Momentum

$$v_{t+1} = \rho v_t - \alpha \nabla f(x_t + \rho v_t)$$
$$x_{t+1} = x_t + v_{t+1}$$

```
dx = compute_gradient(x)
old_v = v
v = rho * v - learning_rate * dx
x += -rho * old_v + (1 + rho) * v
```



- SGD
- SGD+Momentum
- Nesterov

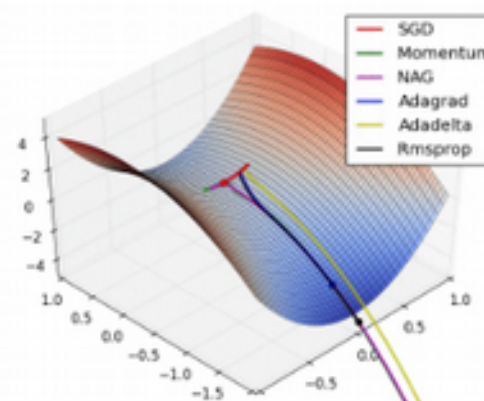
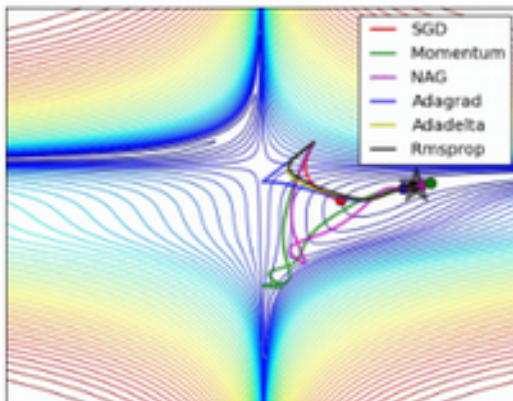
Оптимизаторы: Adam

```
first_moment = 0
second_moment = 0
for t in range(1, num_iterations):
    dx = compute_gradient(x)
    first_moment = beta1 * first_moment + (1 - beta1) * dx
    second_moment = beta2 * second_moment + (1 - beta2) * dx * dx
    first_unbias = first_moment / (1 - beta1 ** t)
    second_unbias = second_moment / (1 - beta2 ** t)
    x -= learning_rate * first_unbias / (np.sqrt(second_unbias) + 1e-7))
```

Momentum

Bias correction

AdaGrad / RMSProp



Функция Потерь

1. Абсолютная

$$L(X_i, W) = |y_i(X, W) - \hat{y}|$$

2. Квадратичная

$$L(X_i, W) = (y_i(X, W) - \hat{y})^2$$

Функция Потерь

3. Кросс – Энтропия – Бинарная \ Binary cross entropy

$$L(X_i, W) = -\frac{1}{N} (\hat{y} \cdot \log(y_i(X, W)) + (1 - \hat{y}) \cdot \log(1 - y_i(X, W)))$$

4. Взвешенная Кросс – Энтропия – Бинарная \ Weighted Binary cross entropy

$$L(X_i, W) = -\frac{1}{N} (\beta \cdot \hat{y} \cdot \log(y_i(X, W)) + (1 - \hat{y}) \cdot \log(1 - y_i(X, W)))$$

5. Balanced binary cross entropy

$$L(X_i, W) = -\frac{1}{N} (\beta \cdot \hat{y} \cdot \log(y_i(X, W)) + (1 - \beta)(1 - \hat{y}) \cdot \log(1 - y_i(X, W)))$$

Функция Потерь

6. Дице коэффициент - Dice index

$$L(X, W) = 2 \frac{|y_i(X, W) \cap \hat{y}_i|}{|y_i(X, W)| + |\hat{y}_i|}$$

7. Jaccard loss - степень сходства

$$L(X_i, W) = \frac{\sum_{i=1, N} y_i(X, W) \cdot \hat{y}_i}{\sum_{i=1, N} y_i(X, W) + \sum_{i=1, N} \hat{y}_i - \sum_{i=1, N} y_i(X, W) \hat{y}_i}$$

Вопросы



GeekBrains



Практическое задание



1. Попробуйте улучшить работу нейронной сети(разобранную на уроке) обучавшейся на датасет Fashion-MNIST.

Опишите в комментарии к уроку - какого результата вы добились от нейросети? Что помогло вам улучшить ее точность?

2. Поработайте с документацией TensorFlow 2. Попробуйте найти полезные команды TensorFlow, не разобранные на уроке.

*3. Попробуйте обучить нейронную сеть на TensorFlow 2 на датасете imdb_reviews.

Опишите в комментарии к уроку - какого результата вы добились от нейросети? Что помогло вам улучшить ее точность?