

# Сверточные сети на процессорах «Эльбрус»

Щадинская А.Н.



# Прикладные библиотеки

Для эффективного использования вычислительной техники пользователю в первую очередь нужны *прикладные библиотеки*, которые упрощают разработку.

The diagram illustrates the development process for two different hardware providers using the PyTorch framework. It is divided into two horizontal sections.

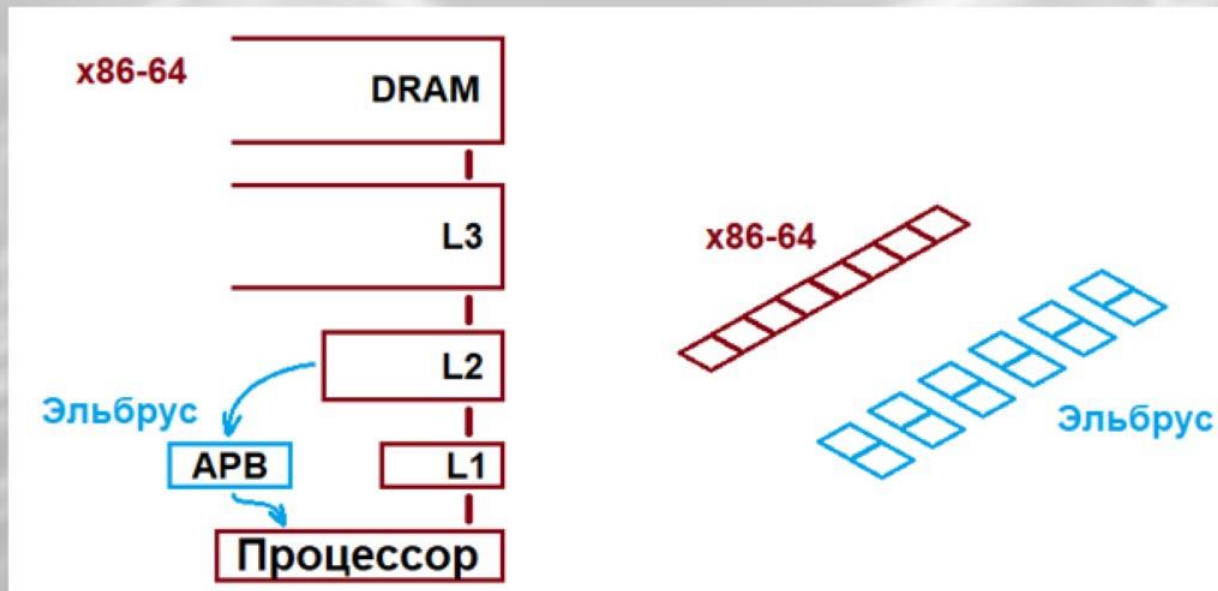
**Top Section (NVIDIA):** On the left, an image of a NVIDIA GPU is labeled "Производитель" (Manufacturer). In the center, a box contains the "cuDNN" logo. On the right, the PyTorch logo is labeled "Пользователь" (User). A green oval encircles the cuDNN logo, and a blue oval encircles the PyTorch logo. These two ovals overlap in the center, with a blue line connecting the manufacturer to the user.

**Bottom Section (Intel):** On the left, an image of an Intel Movidius Myriad 2 VPU is labeled "Производитель" (Manufacturer). In the center, a large black question mark is shown. On the right, the PyTorch logo is labeled "Пользователь" (User). A green oval encircles the question mark, and a blue oval encircles the PyTorch logo. These two ovals overlap in the center, with a blue line connecting the manufacturer to the user.

**Right Image:** A street scene with a camera overlay. The overlay shows a green lane on the road, yellow bounding boxes around a white van and a black car, and red bounding boxes around various traffic signs. A data panel in the top right corner is titled "CAMERA\_01\_2015" and contains technical details.

# Особенности архитектуры «Эльбрус»

- Наличие асинхронного буфера, работающего одновременно с традиционной кэш памятью L1.
- 6 векторных команд за такт (вместо 1 – 2 в x86-64).
- Более сложный язык ассемблера.



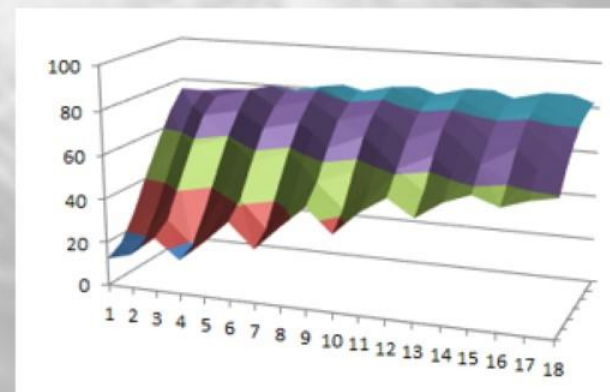
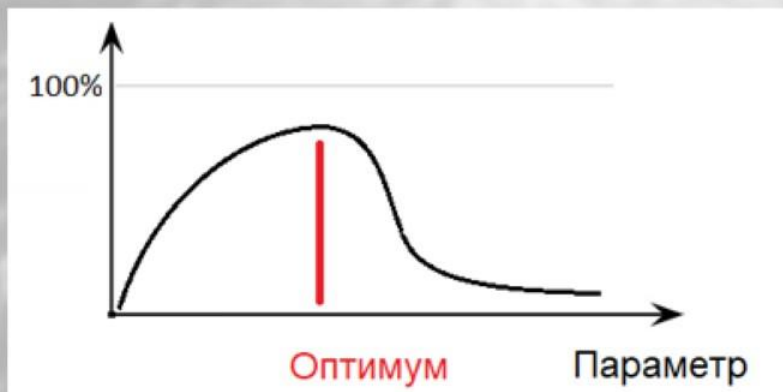
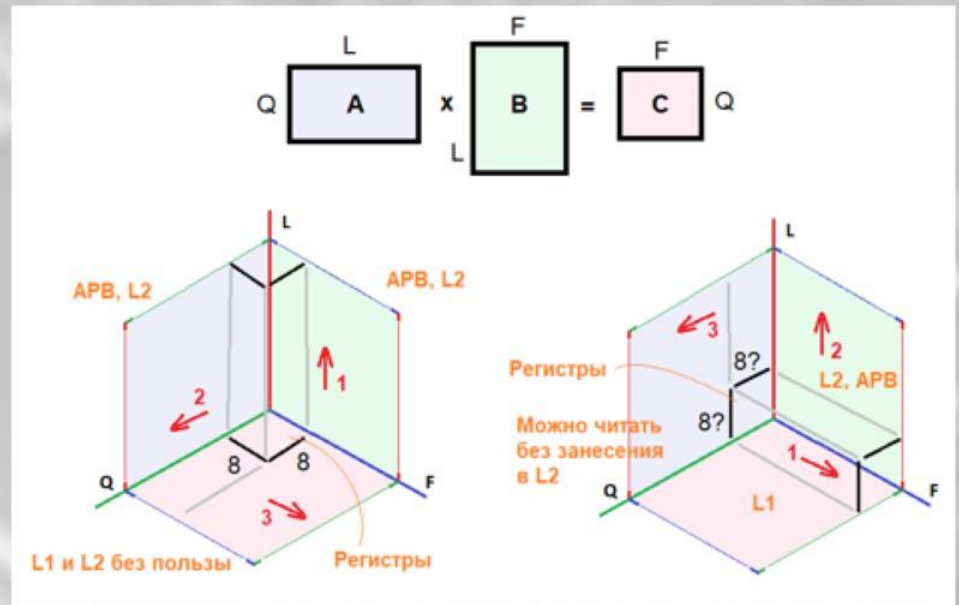


# Приемы программирования

```

1268 #pragma unroll(2)
1269 for(int il = 0; il < rxL; il += 2) {
1270
1271     a0 = *((uint64b *) (a +      iryxL + il));
1272     b0 = *((uint64b *) (b +      iryrxL + il));
1273     b1 = *((uint64b *) (b +      ryrxL + iryrxL + il));
1274     b2 = *((uint64b *) (b + 2 * ryrxL + iryrxL + il));
1275     b3 = *((uint64b *) (b + 3 * ryrxL + iryrxL + il));
1276     b4 = *((uint64b *) (b + 4 * ryrxL + iryrxL + il));
1277     b5 = *((uint64b *) (b + 5 * ryrxL + iryrxL + il));
1278     b6 = *((uint64b *) (b + 6 * ryrxL + iryrxL + il));
1279     b7 = *((uint64b *) (b + 7 * ryrxL + iryrxL + il));
1280
1281     c0 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b0), c0);
1282     c1 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b1), c1);
1283     c2 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b2), c2);
1284     c3 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b3), c3);
1285     c4 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b4), c4);
1286     c5 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b5), c5);
1287     c48 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b6), c48);
1288     c49 = __builtin_e2k_pfmuls(__builtin_e2k_pfmuls(a0, b7), c49);
1289 }
1290 *((unsigned *) c) = __builtin_e2k_pfmuls(c0, c0);
1291 *((unsigned *) c + 1) = __builtin_e2k_pfmuls(c1, c1);
1292 *((unsigned *) c + 2) = __builtin_e2k_pfmuls(c2, c2);
1293 *((unsigned *) c + 3) = __builtin_e2k_pfmuls(c3, c3);
1294 *((unsigned *) c + 4) = __builtin_e2k_pfmuls(c4, c4);
1295 *((unsigned *) c + 5) = __builtin_e2k_pfmuls(c5, c5);
1296 *((unsigned *) c + 6) = __builtin_e2k_pfmuls(c48, c48);
1297 *((unsigned *) c + 7) = __builtin_e2k_pfmuls(c49, c49);

```



# Пример результатов

С помощью детального изучения микроархитектуры «Эльбрус» и учета специфики сверточных нейронных сетей удалось реализовать функции, по своей эффективности не уступающие функциям, реализованным МЦСТ, или даже *превосходящие*.

На примере умножения матриц на 8 ядрах процессора Эльбрус-8С:

	<b>«Платформа» (ГосНИИАС)</b>	<b>Библиотека EML (МЦСТ)</b>
1024x1024x1024	67.5%	53.9%
2048x2048x2048	80.25%	68.25%
4016x4016x4088	82.25%	61.37%

Такой уровень результатов позволяет ГосНИИАС *гарантировать* большую скорость работы сверточных нейронных сетей на «Эльбрусах».



# Текущие работы

«Платформа» – отечественный фреймворк для работы со сверточными нейронными сетями.

«Супер-ЭВМ-ППО» – создание ПО для обучения сверточных нейронных сетей на суперкомпьютере на базе «Эльбрус-8СВ».

