

Рис. 0.1.

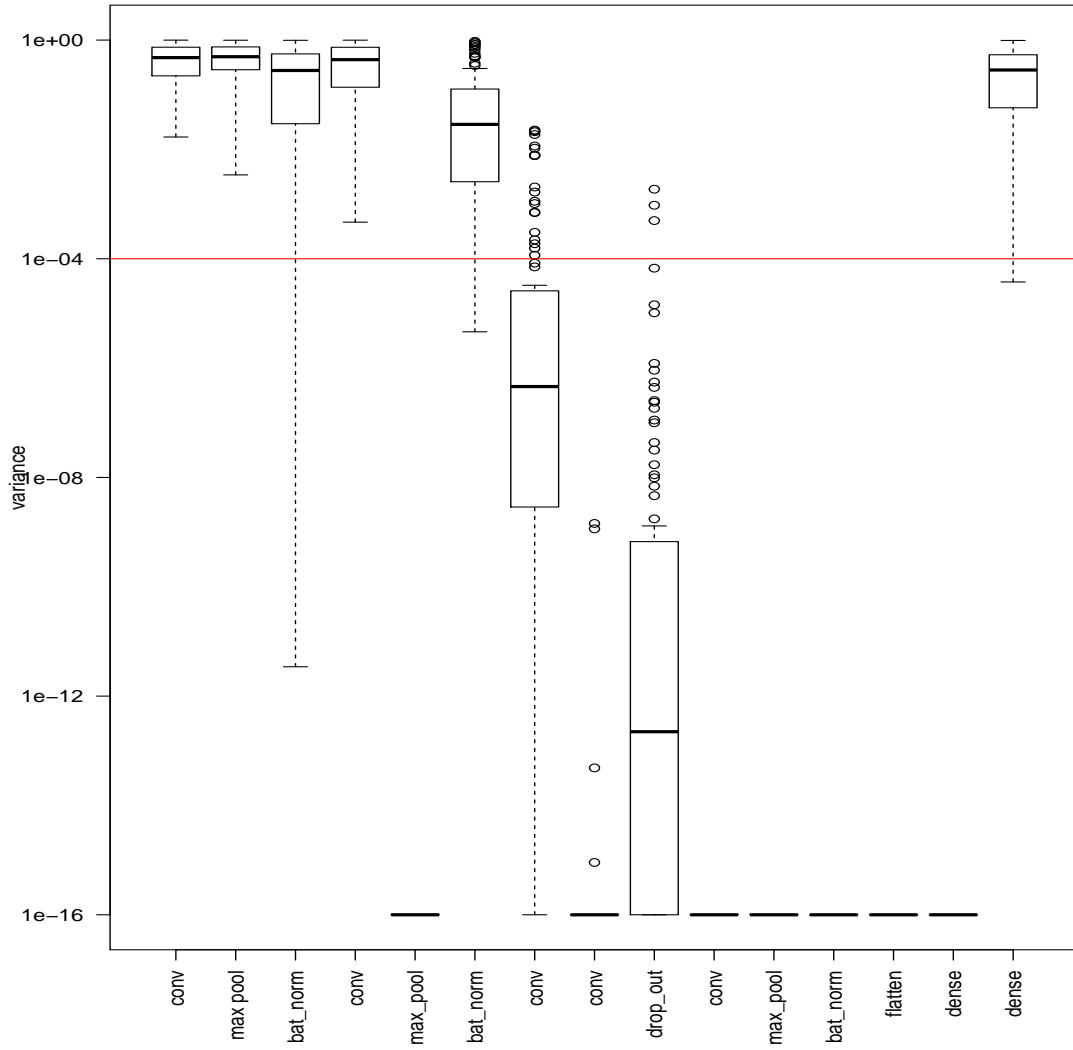


Рис. 0.2.

Посчитаем несмещенные оценки математического ожидания и дисперсии для выхода каждого слоя (для 10000 образцов). Первое число в нижнем индексе обозначает слой, второе результат, который ожидаем на выходе (0 или 1).

$$m_{0_0} = 3.9753 \quad m_{0_1} = 3.9817 \quad var_{0_0} = 146.9832 \quad var_{0_1} = 147.5937$$

$$m_{1_0} = 6.0533 \quad m_{1_1} = 6.0630 \quad var_{1_0} = 251.6746 \quad var_{1_1} = 252.6236$$

$$m_{2_0} = 0.0087 \quad m_{2_1} = 0.01034 \quad var_{2_0} = 1.5715 \quad var_{2_1} = 1.5868$$

$$m_{3_0} = 7.1735 \quad m_{3_1} = 7.1674 \quad var_{3_0} = 137.2314 \quad var_{3_1} = 137.4925$$

$$m_{4_0} = 13.2688 \quad m_{4_1} = 13.2709 \quad var_{4_0} = 218.4356 \quad var_{4_1} = 219.5417$$

$$m_{5_0} = -0.0025 \quad m_{5_1} = -0.0014 \quad var_{5_0} = 0.8465 \quad var_{5_1} = 0.8580$$

$$m_{6_0} = 0.0680 \quad m_{6_1} = 0.0724 \quad var_{6_0} = 0.1617 \quad var_{6_1} = 0.1858$$

$$m_{7_0} = 0.7459 \quad m_{7_1} = 0.7880 \quad var_{7_0} = 4.8757 \quad var_{7_1} = 5.6263$$

$$m_{8_0} = 0.7465 \quad m_{8_1} = 0.7880 \quad var_{8_0} = 8.5122 \quad var_{8_1} = 9.7856$$

$$m_{9_0} = 11.6431 \quad m_{9_1} = 13.5871 \quad var_{9_0} = 967.1589 \quad var_{9_1} = 1270.1080$$

$$m_{10_0} = 37.0520 \quad m_{10_1} = 43.4176 \quad var_{10_0} = 2509.943 \quad var_{10_1} = 3255.228$$

$$m_{11_0} = 0.0659 \quad m_{11_1} = 0.1120 \quad var_{11_0} = 0.1420 \quad var_{11_1} = 0.1981$$

$$m_{12_0} = 0.0659 \quad m_{12_1} = 0.1120 \quad var_{12_0} = 0.1420 \quad var_{12_1} = 0.1981$$

$$m_{13_0} = 0.3778 \quad m_{13_1} = 0.2505 \quad var_{13_0} = 1.6201 \quad var_{13_1} = 2.0115$$

$$m_{14_0} = 0.2122 \quad m_{14_1} = 0.7955 \quad var_{14_0} = 0.0866 \quad var_{14_1} = 0.0545$$

Исходя из этих статистических моментов, можно отметить, что последний batch normalization ведет себя очень странно (дисперсия слишком мала). Возможно это связано с большим количеством нулей на входе в него. Причем нули, которые получились до входа в последний batch norm, этот слой превращает в какую-то другую ненулевую константу, а слой, который следует за ним опять обнуляет большинство из них. Поэтому можно сделать вывод, что последний batch norm несет больше вреда, чем пользы и от него можно избавиться. Поэтому я прогнал эту же сетку без последнего batch norm. Но, к сожалению, результаты получились такими же как и до этого (может быть чуть-чуть лучше). Еще забавно, что dropout увеличил дисперсию, а не уменьшил как я предполагал.

Насчет картинок, которые получились после понейронного анализа. На первых трех слоях сетке не особо важно ноль в результате получается или один. Далее на второй свертке и втором пулинге появляются признаки того, что сетка начинает различать ноль и единицу, но потом batch norm немного портит картину. В общем насчет batch norm у меня есть сомнения. Первый точно нужен, а остальные два как будто мешают. Но я поигрался с сеткой и у меня не получилось добиться лучшего результата, чем был. Далее опять сетка пытается разделять ноль и один и после пятой свертки можно сказать, что она определилась с выбором (в том плане, что распределения становятся разными). Тем самым можно предположить, что слои, которые после 5 свертки не особо и нужны. Возможно, имеет смысл так же исследовать выборочные моменты по нейронно, это дополнит картину.

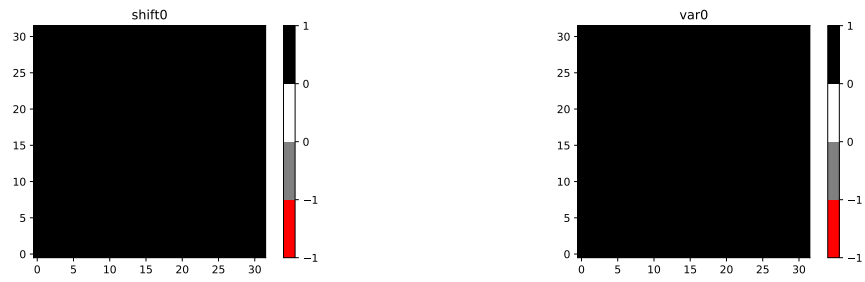


Рис. 0.3. conv 1

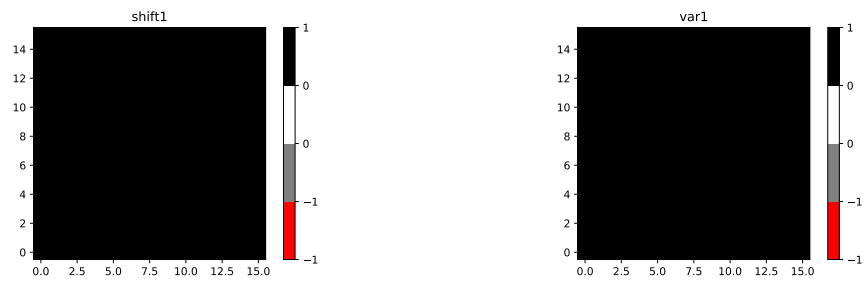


Рис. 0.4. max pool 1

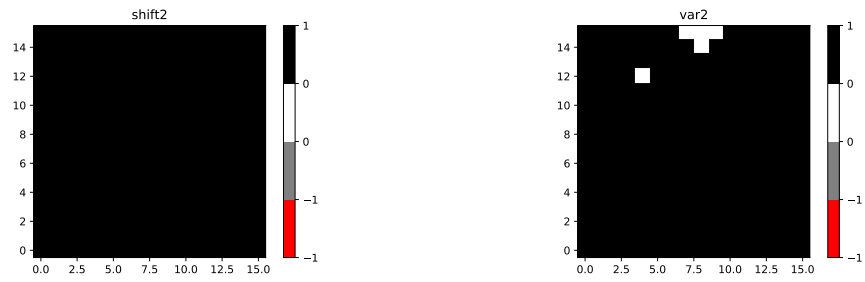


Рис. 0.5. batch norm 1

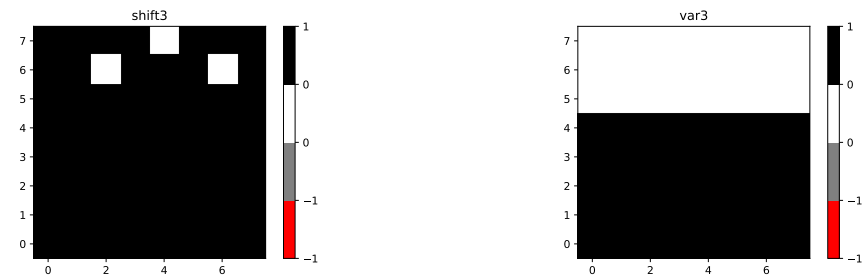


Рис. 0.6. conv 2

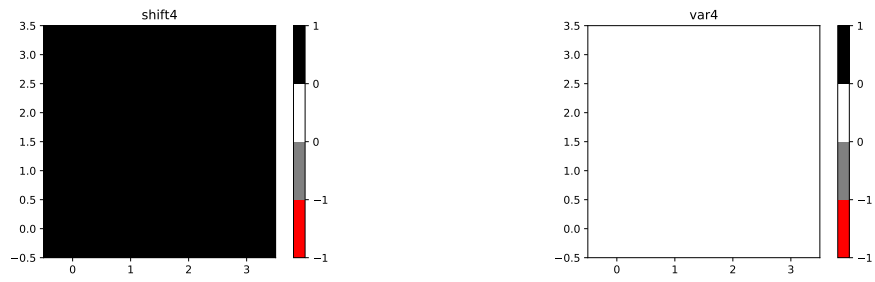


Рис. 0.7. max pool 2

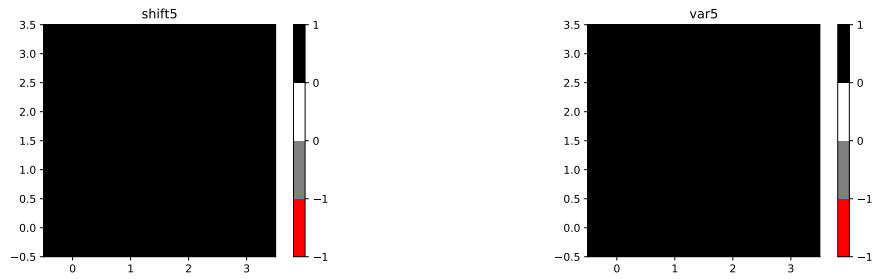


Рис. 0.8. batch norm 2

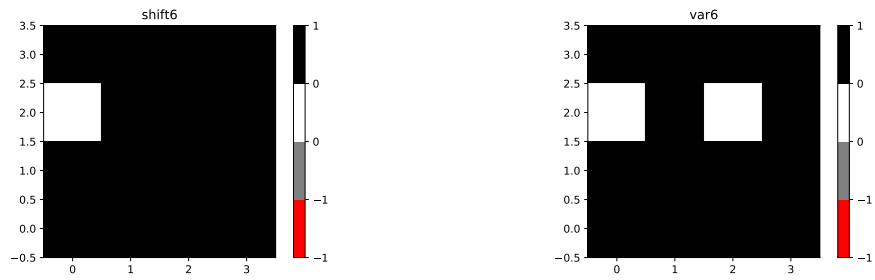


Рис. 0.9. conv 3

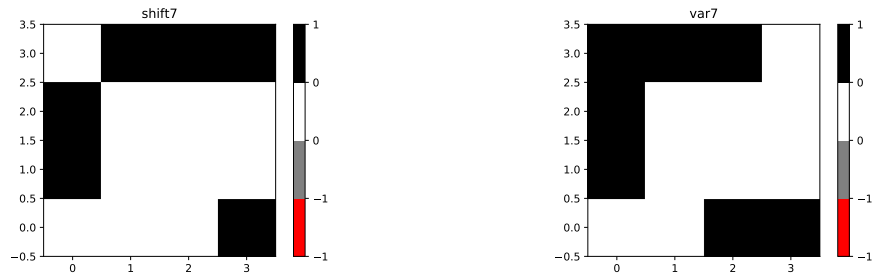


Рис. 0.10. conv 4

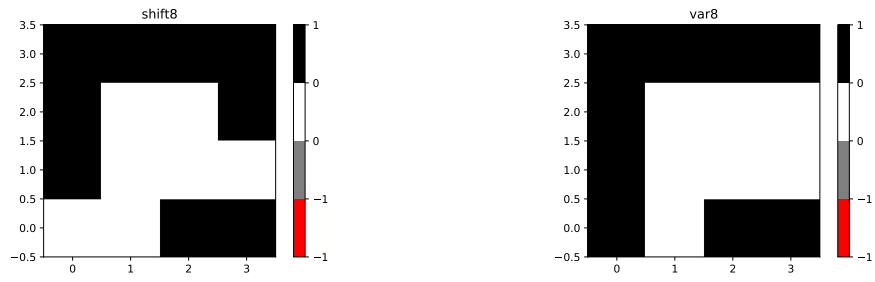


Рис. 0.11. dropout 1

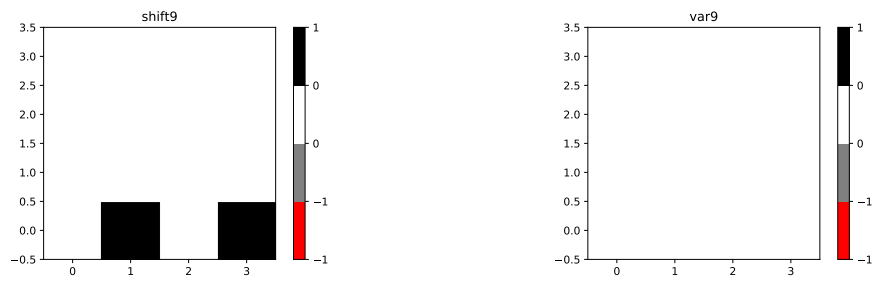


Рис. 0.12. conv 5

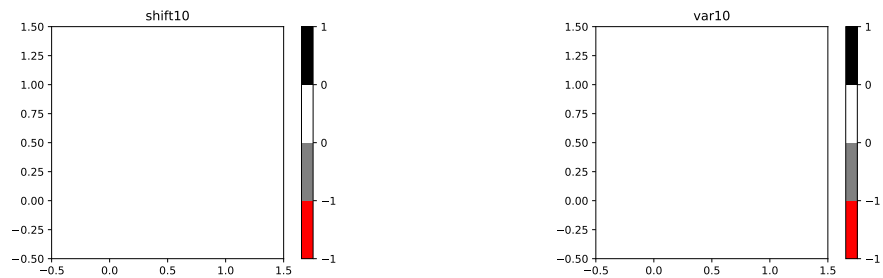


Рис. 0.13. max pool 3

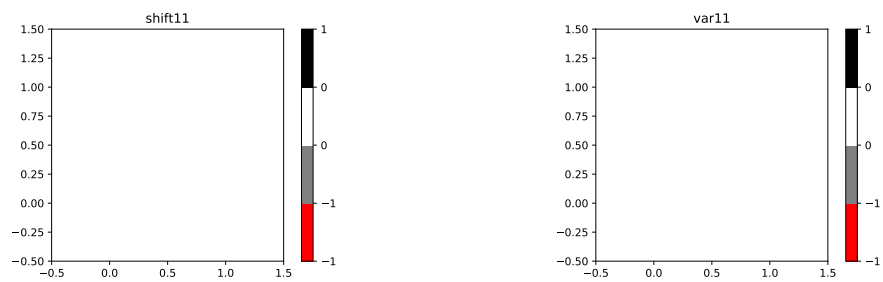


Рис. 0.14. batch norm 3

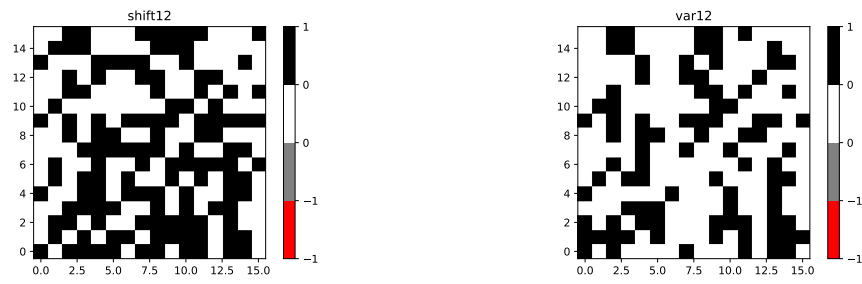


Рис. 0.15. flatten 1



Рис. 0.16. dense 1