

Wyniki

Bartosz Rzepkowski

March 2020

Legenda

W trakcie pracy nad tymi wynikami zmieniłem sposób generowania zbioru uczącego sieci. Wcześniej procedura generowania takiego zbioru wyglądała następująco:

- wygeneruj losowy, znormalizowany wektor $|\psi_0\rangle$,
- znajdź wektor poddany bardzo krótkiej ewolucji czasowej (np. 0.1 jednostek czasu) $|\psi_t\rangle$,
- dodaj oba wektory do zbioru uczącego (pierwszy jako wektor wejściowy a drugi jako oczekiwany wyjściowy).

Teraz generowanie tego zbioru wygląda następująco:

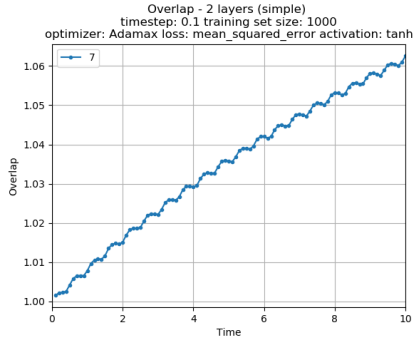
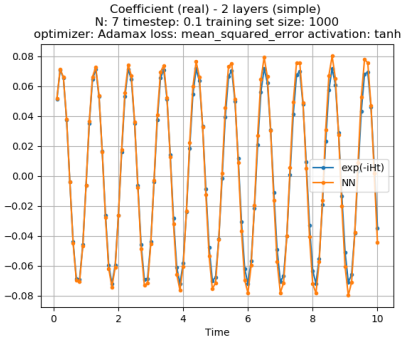
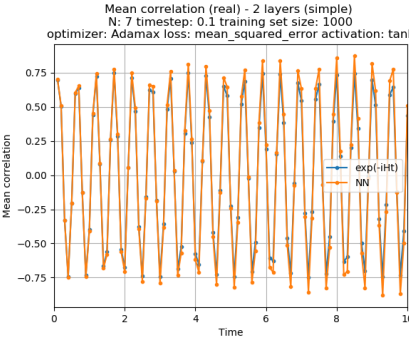
- wygeneruj losowy, znormalizowany wektor $|\psi_0\rangle$,
- znajdź wektor poddany bardzo krótkiej ewolucji czasowej (np. 0.1 jednostek czasu) $|\psi_t\rangle$,
- dodaj oba wektory do zbioru uczącego (pierwszy jako wektor wejściowy a drugi jako oczekiwany wyjściowy),
- zamiast generować kolejny wektor losowy przeprowadź krótką ewolucję wektora $|\psi_t\rangle$ uzyskując wektor $|\psi_{2t}\rangle$ i ponownie dodaj te dwa wektory do zbioru uczącego,
- powtórz powyższy krok aż do momentu uzyskania całkowitego porządanego czasu ewolucji. Tzn. np. dla kroku (t) długości 0.1 i czasu całkowitego (t_{total}) równego 10 jednostek czasu wykonałobyśmy 100 takich operacji: $|\psi_0\rangle, |\psi_t\rangle, |\psi_{2t}\rangle, \dots, |\psi_{100t}\rangle = |\psi_{t_{total}}\rangle$,
- dopiero po zakończeniu opwyższej procedury wygeneruj kolejny wektor losowy.

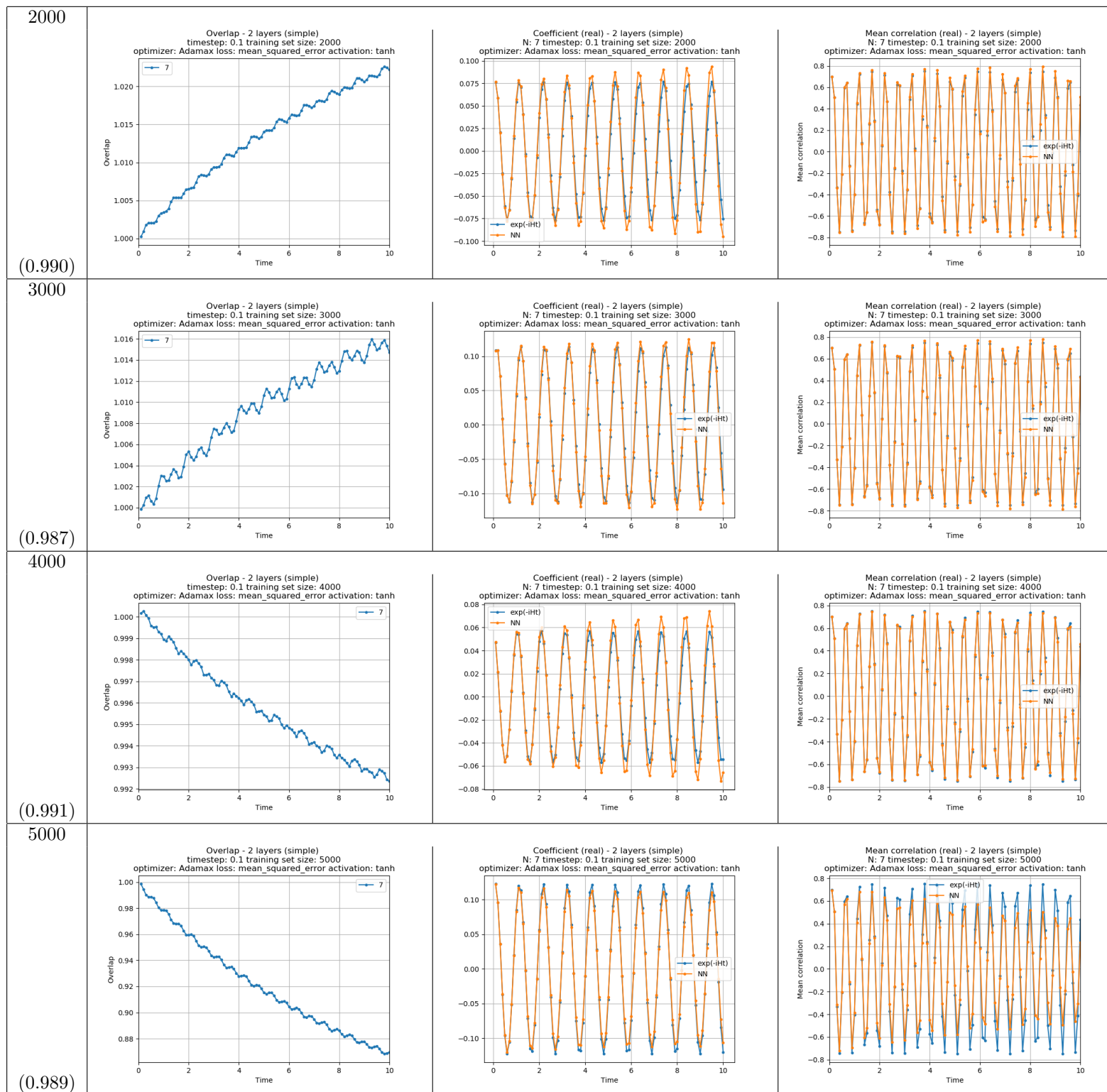
Na samym końcu w obu przypadkach wykonywałem "przetrasowanie" zbiorów uczących. Czyli w najnowszej wersji sieć nie uczyła się na występujących bezpośrednio po sobie kolejnych krokach ewolucji, tylko raczej na losowo wybranych jej momentach.

1 Szukanie dobrego rozmiaru zbioru uczącego (2 warstwy, Adamax, mean squared error, tanh)

W tej sekcji szukałem najlepszego możliwego rozmiaru zbioru uczącego dla nowego sposobu uczenia sieci. Oznaczenie "training_set_size: 1000" oznacza wygenerowanie 1000 wektorów losowych. Jednak docelowo cały zbiór uczący ma 100 000 elementów, ponieważ dla każdego losowego wektora do zbioru dodajemy 100 kolejnych kroków jego ewolucji.

Wlewej kolumnie oznaczonej jako # zawarto liczbę wektorów losowych generowanych w trakcie tworzenia zbioru uczącego. Wartość w nawiasie w tej kolumnie oznacza precyzję sieci dla JEDNEGO kroku ewolucji (czyli w naszym przypadku dla kroku = 0.1).

#	Overlap	Korelacja	Współczynnik
1000 (0.992)			



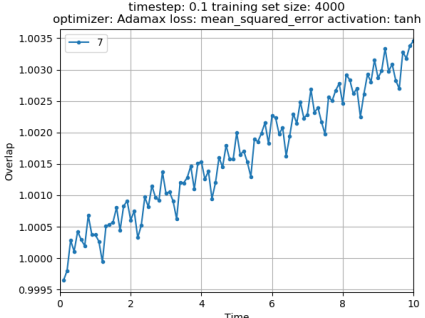
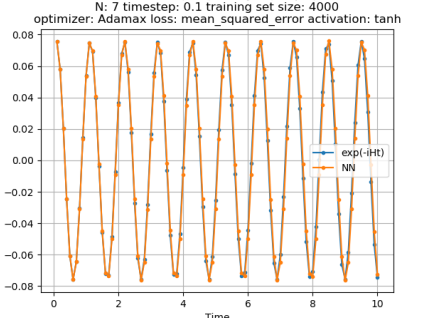
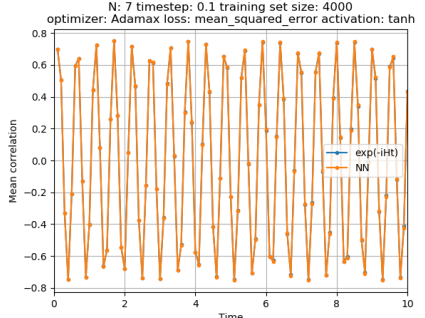
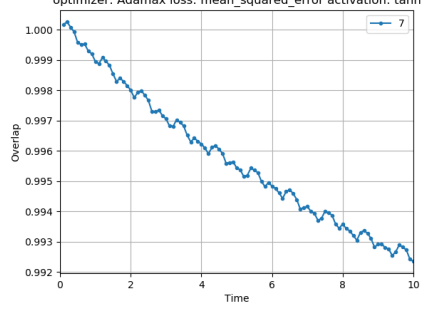
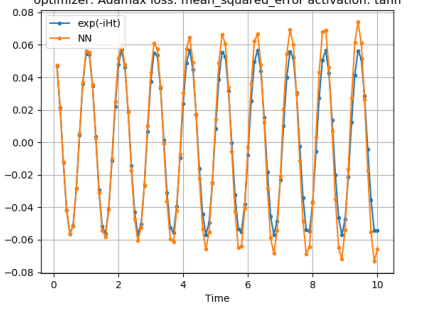
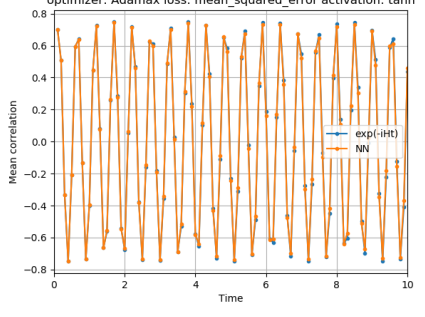
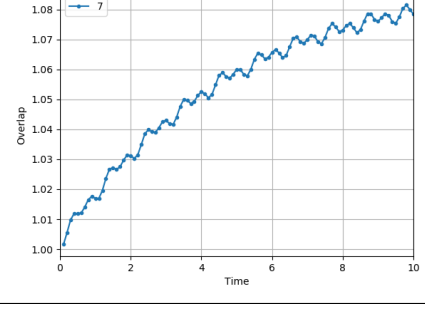
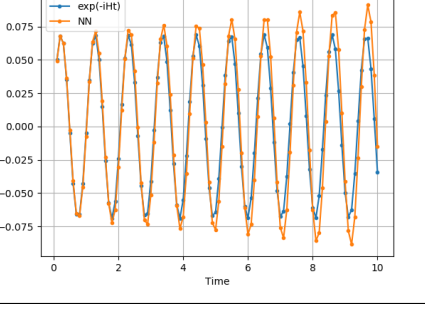
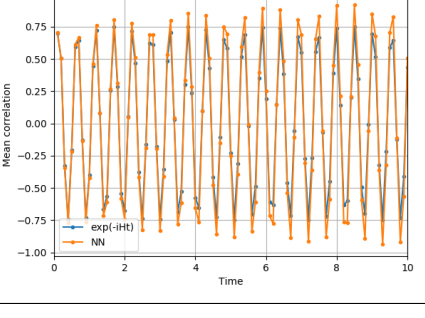
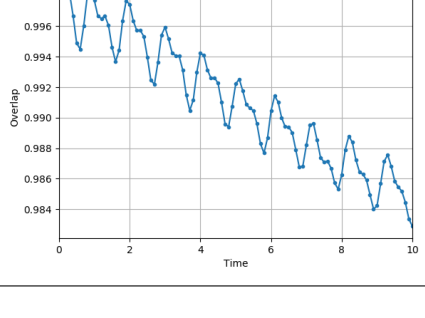
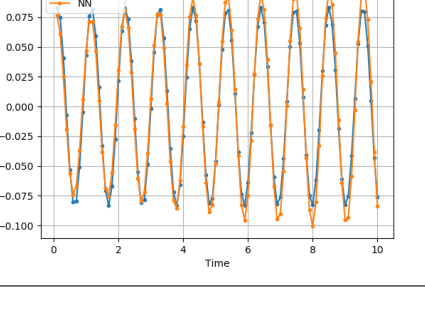
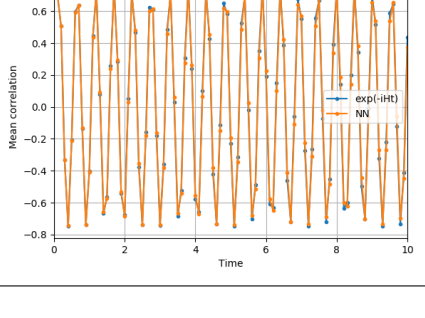
1.1 Wnioski do tej sekcji

- Najlepsza precyzja występowała dla zbioru uczącego rozmiaru 400 000 (4000 losowych wektorów poddanych ewolucji). Wtedy tracimy na dokładności jedynie 0.008 dla całkowitego docelowego czasu ewolucji!
- Ciekawe jest to, że wystarczą 4000 losowe wektory, które są tylko poddawane ewolucji. Ponadto, takie wyniki są dużo lepsze niż dla 1 000 000 losowo wygenerowanych wektorów!
- Mimo "przetrasowania" zbioru uczącego, wciąż dostajemy dobre rezultaty.
- Sieć dąży do takiej sytuacji, w której współczynnik będzie oscylował między około 0.1 a -0.1. Widać to najlepiej w przypadku z 4000 losowymi wektorami, tzn. późniejsze wartości współczynników są coraz bardziej "rozstrzelone". Miałem przykład, na którym było to lepiej widać, jednak pochopnie go usunąłem myśląc, że sieć się źle nauczyła...

2 Sprawdzenie dokładności dla większej liczby warstw

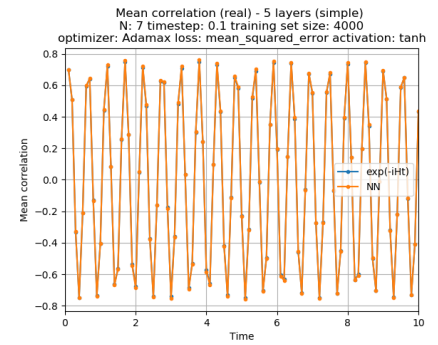
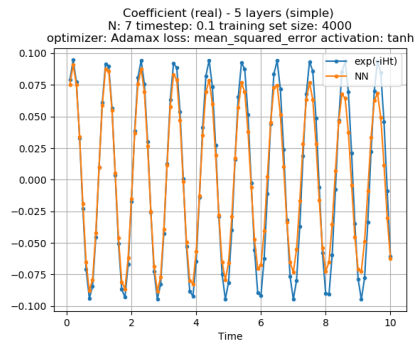
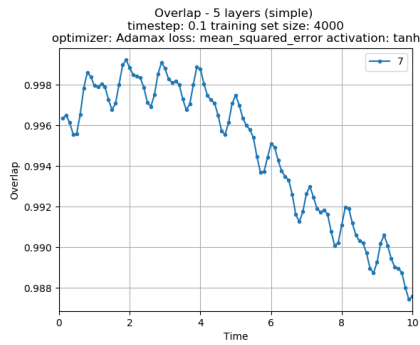
W tej sekcji chciałem sprawdzić, jak będzie sobie radziła sieć z takimi samymi parametrami jak ta z poprzedniej sekcji (Adamax, mean squared error, tanh), ale dla większej liczby warstw.

Tym razem w kolumnie z lewej oznaczonej jako # podana jest liczba warstw w sieci (wszystkie wersje sieci uczone są na zbiorze uczącym złożonym z 400 000 wektorów, czyli było wygenerowanych 4000 wektorów losowych).

#	Overlap	Korelacja	Współczynnik
1 (0.994)	<p>Overlap - 1 layer (simple) timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Coefficient (real) - 1 layer (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Mean correlation (real) - 1 layer (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 
2 (0.991)	<p>Overlap - 2 layers (simple) timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Coefficient (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Mean correlation (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 
3 (0.988)	<p>Overlap - 3 layers (simple) timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Coefficient (real) - 3 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Mean correlation (real) - 3 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 
4 (0.925)	<p>Overlap - 4 layers (simple) timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Coefficient (real) - 4 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 	<p>Mean correlation (real) - 4 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</p> 

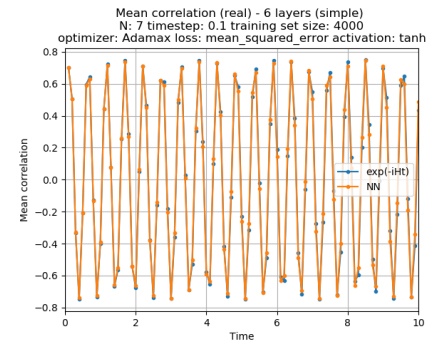
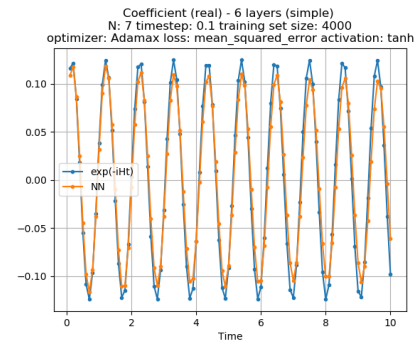
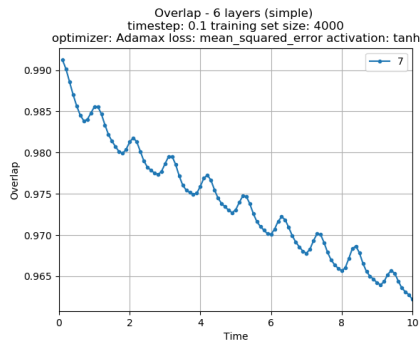
5

(0.823)



6

(0.738)



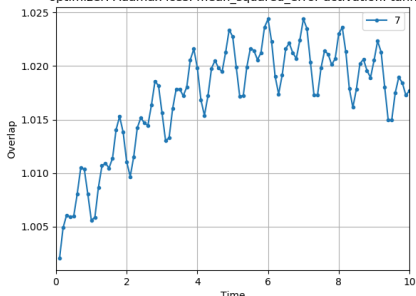
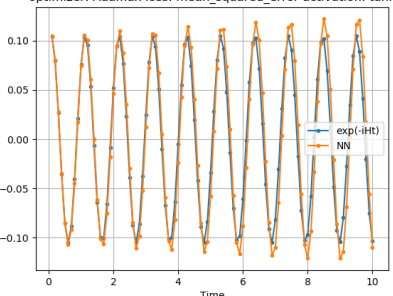
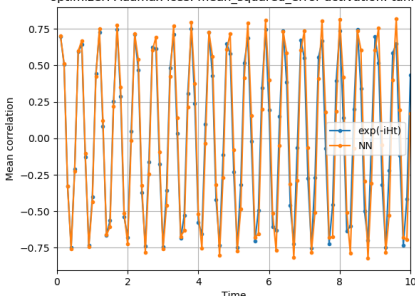
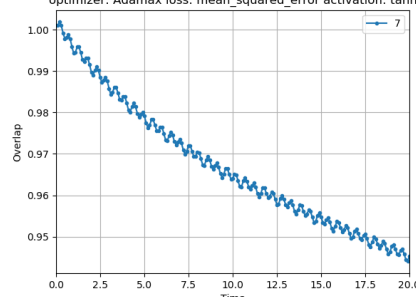
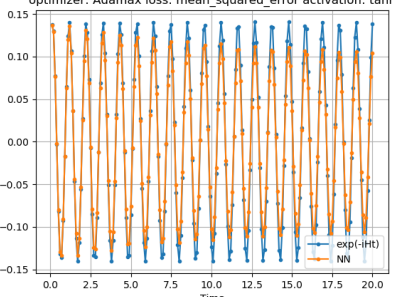
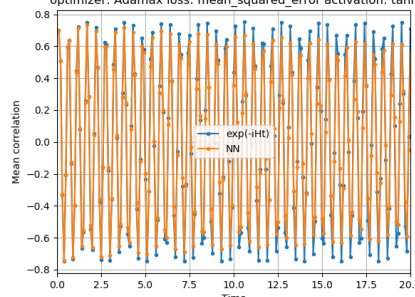
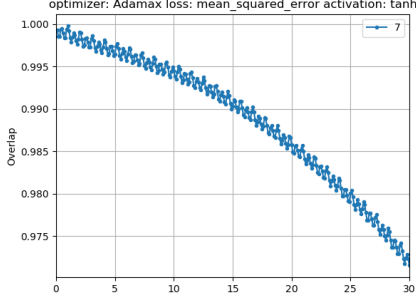
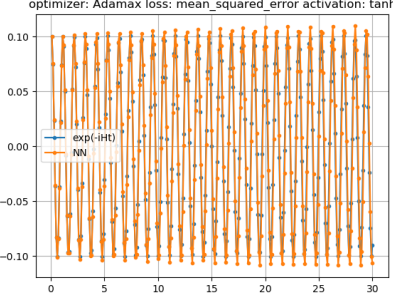
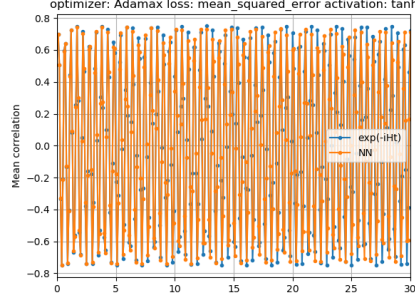
2.1 Wnioski do tej sekcji:

- Co ciekawe, ciężko znaleźć bezpośrednie powiązanie między dokładnością sieci dla jednego kroku ewolucji a jej zachowaniem w dłuższej perspektywie. Tzn. np. sieć z 6 warstwami dla jednego kroku czasu miała dokładność wynoszącą ok. 74%, ale overlap już dla tesowanej ewolucji utrzymywał się na poziomie powyżej 95%.

3 Zbadanie zachowania dla sieci z dwoma warstwami, ale dłuższym całkowitym czasem ewolucji w trakcie ostatecznego testowania

W tej sekcji chciałem sprawdzić, jak sieć ucząca się na wektorach dla całkowitej ewolucji wynoszącej maksymalnie 10 jednostek czasu poradzi sobie w dłuższej perspektywie, tzn. np. dla całkowitej ewolucji wynoszącej 20 i 30 jednostek.

Tym razem w lewej kolumnie oznaczonej jako # podano mnożnik, przez jaki przemnożono docelowy całkowity czas ewolucji. Czyli w pierwszym wierszu są wyniki dla noramlnnej ewolucji (10 jednostek czasu), w drugim dla 20 jednostek, a w 3 dla 30 jednostek.

#	Overlap	Korelacja	Współczynnik
1x	<div>Overlap - 2 layers (simple) timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 	<div>Coefficient (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 	<div>Mean correlation (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 
(0.989)			
2x	<div>Overlap - 2 layers (simple) timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 	<div>Coefficient (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 	<div>Mean correlation (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 
(0.991)			
3x	<div>Overlap - 2 layers (simple) timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 	<div>Coefficient (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 	<div>Mean correlation (real) - 2 layers (simple) N: 7 timestep: 0.1 training set size: 4000 optimizer: Adamax loss: mean_squared_error activation: tanh</div> 
(0.991)			

3.1 Wnioski do tej sekcji

- Sieć dość dobrze sobie radzi też dla dłuższych czasów ewolucji, niż te przewidziane w zbiorze uczącym.