the same non-replicated units as in (a) appear, and the two units that were replicated in (a) appear six times in (b) but with smaller weight magnitudes. This is the scaling we were talking about before. When the number of hidden units is very large, the weight magnitudes of the replicated units become practically null, but they keep globally forming the same linear mapping.

We approximate experimentally the functional distance as the mean square distance between the outputs of two networks in a grid of 10,000 points regularly distributed in the input domain. Figure 3 shows the functional distances between architectures with different number of hidden units, minimized for several weight-decay coefficient values α. Since for some α's, some of the networks fell in local minima, in these cases we used the best minimum selected from three different random trials. It is evident that as α grows, all the architectures tend to produce the same results. But the most interesting observation from this graph is that, for any $\alpha > 0$, the distances between architectures decrease very quickly as the number of hidden units grows, and are indistinguishable from zero above 50 units. Notice that the comparisons involve networks that differ the more in the number of hidden units, the larger are the architectures. The above observation agrees with the expectation of a tendency to closer similarity for larger nets. Of course all the architectures exhibit almost the same generalization error for any positive α, especially those above 50 HU's.

Discussion

In this paper we have put forth a definition of functional invariance, which basically states that a learning method is functionally invariant if, when applied to increasingly large networks, the output for every possible input tends to a limit, and have examined what kinds of methods possess this property. We have identified three mechanisms that can originate functional invariance:

- bayesian treatment of neural networks with weight priors that converge to a prior over functions,
- implicit definition of a mean target for the complete input space, and
- additively decomposable regularizers that produce minima with a finite number of nonlinear units in the limit of infinite units.

Examples of the first two types of mechanisms are bayesian learning with Neal's type weight priors, and input noise addition using probability density functions taking non-zero values in all their domain, respectively. Examples of the third type are the regularizer that emulates learning with random weights and classical weight-decay. There are relations between these mechanisms, but it is difficult to see a unifying principle. For example, the second type can be viewed as defining a prior over input-output functions, as the first type does, namely one that always concentrates the probability on a single function. However, this prior is the same for all networks using the second type of mechanism, while in the first mechanism the prior over functions is approximated only for large networks. In addition, since the target function is completely defined in the second type of mechanism, the distance to that function is

directly minimized, whereas in the first type, averaging over the probability distributions is required to guarantee functional invariance.

There are also differences in the type of units that these mechanisms generate. The third type produces only a finite number of nonlinear units in the infinite limit, while the second gives rise to an infinite number of them. Take into account that the implicit target function can be anyone and, therefore, an infinite number of nonlinear units is required to approximate it [2].

It could seem strange that no one (up to our knowledge) had observed the functional invariance of, for example, weight-decay. However, careful optimizations are required to observe regular patterns in the weight configurations and thus see this property with sharpness. This does not mean that these results are not of practical relevance; as two different but large networks are brought moderately close to a global minimum, the functional distance between them becomes very low. The problem of falling into local minima can be significant, but their frequency using weight-decay is apparently not high. For example, in the experiment of Figure 3 comparing the functional distances of several nets with different weight-decay coefficients, among the numerous optimizations required, only three times we got a local minimum in a single trial.

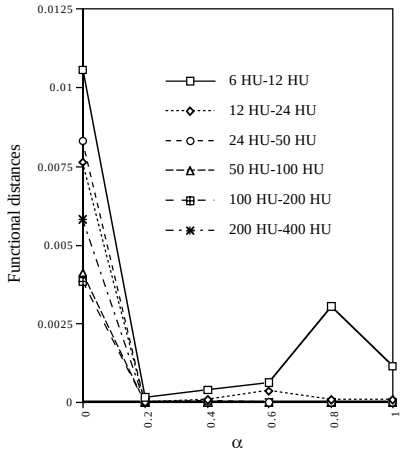

Fig. 3. Evaluation of the similarity between the functions implemented by architectures with different numbers of hidden units. Values spanning a wide range of the weight-decay coefficient α are tested.

References

1. Bishop, C.M.: "Neural networks for pattern recognition". Oxford University Press, 1995.
2. Hornik, K.: "Approximation capabilities of multilayer feedforward networks". *Neural Networks*, Vol 4 (2), pp. 251-257, 1991.
3. Koistinen, P. and Holmstrom, L.: "Kernel regression and backpropagation training with noise", Advances in Neural Information Processing Systems 4, Morgan-Kauffman, 1992.
4. Neal, R.M.: "Bayesian learning for neural networks". Springer-Verlag, New York, 1996.
5. Ruiz de Angulo, V. and Torras, C.: “Random weights and regularization”, ICANN´94, Sorrento, pp. 1456-1459, 1994.
6. Ruiz de Angulo, V. and Torras, C.: “A deterministic algorithm that emulates learning with random weights”, *Neurocomputing* (to appear).
7. Ruiz de Angulo, V. and Torras, C.: "Architecture-independent approximation of functions", *Neural Computation*, Vol. 13, No. 5, pp. 1119-1135, May 2001.
8. Wolpert, D.H. (1994): "Bayesian backpropagation over I-O function rather than weights", Advances in Neural Information Processing Systems 6, Morgan Kauffman, 1999.